



TESIS - KI142502

**PENINGKATAN KINERJA *PREDICTION ERROR EXPANSION*  
DALAM *DATA HIDING* DENGAN MEREDUKSI *ERROR EXPANSION*  
DAN MENGELOMPOKKAN NILAI PIKSEL SECARA ADAPTIF**

Hendro Eko Prabowo  
NRP. 5116201006

DOSEN PEMBIMBING

Tohari Ahmad, S.Kom., MIT., Ph.D.  
NIP: 197505252003121002

PROGRAM MAGISTER

RUMPUN MATA KULIAH KOMPUTASI BERBASIS JARINGAN

DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA, 2018

*[Halaman ini sengaja dikosongkan]*

## LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar  
Magister Komputer (M.Kom)  
di  
Institut Teknologi Sepuluh Nopember Surabaya

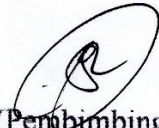
oleh:  
HENDRO EKO PRABOWO  
NRP.5116201006

Dengan judul:  
Peningkatan Kinerja *Prediction Error Expansion* dalam Data Hiding dengan  
Mereduksi *Error Expansion* dan Mengelompokkan Nilai Piksel Secara Adaptif

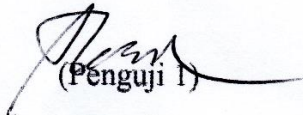
Tanggal Ujian : 4-1-2018  
Periode Wisuda : 2018 Gasal

Disetujui oleh:

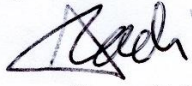
Tohari Ahmad, S.Kom., M.IT., Ph.D.  
NIP. 197505252003121002

  
(Pembimbing 1)

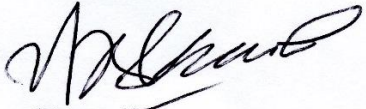
Prof. Ir. Supeno Djanali, M.Sc., Ph.D.  
NIP. 194806191973011001

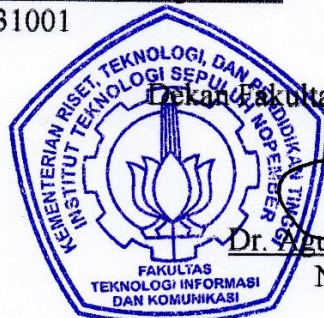
  
(Penguji 1)

Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.  
NIP. 1984101620081210002

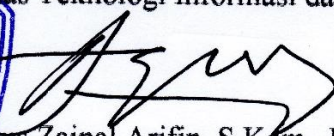
  
(Penguji 2)

Waskitho Wibisono, S.Kom., M.Eng., Ph.D.  
NIP. 197410222000031001

  
(Penguji 3)



Dekan Fakultas Teknologi Informasi dan Komunikasi,

  
Dr. Agus Zainal Arifin, S.Kom., M.Kom.  
NIP. 197208091995121001

*[Halaman ini sengaja dikosongkan]*

# **Peningkatan Kinerja *Prediction Error Expansion* dalam Data Hiding dengan Mereduksi *Error Expansion* dan Mengelompokkan Nilai Piksel Secara Adaptif**

Nama Mahasiswa : Hendro Eko Prabowo  
NRP : 5116 201 006  
Pembimbing : Tohari Ahmad, S.Kom., M.IT., Ph.D.

## **ABSTRAK**

Pertukaran informasi saat ini semakin mudah karena perkembangan teknologi informasi dan komunikasi yang semakin pesat. Kondisi ini terbukti dengan meningkatnya penggunaan IP global sebesar 88,7 EB (*exabytes*) perbulan pada tahun 2016. Peningkatan tersebut menjadikan keamanan informasi sebagai salah satu kebutuhan terpenting untuk melindungi data-data sensitif misalnya data finansial perusahaan atau data keamanan negara. Keamanan dari data sensitif dapat diperoleh menggunakan metode bernama *data hiding*. Teknik ini melindungi data dengan menyisipkannya kedalam sebuah objek (*carrier*) seperti citra digital. Algoritma matematis yang digunakan pada *data hiding* akan memodifikasi piksel pada *carrier* sehingga mendapatkan nilai baru. Modifikasi yang dilakukan sebaiknya menjaga agar *carrier* tidak berubah terlalu signifikan dan dapat menyediakan kapasitas penyembunyian data yang besar. Kebutuhan ini yang mendorong algoritma matematis pada *data hiding* terus dikembangkan.

Salah satu algoritma matematis untuk *data hiding* adalah *prediction error expansion* (PEE). Metode ini menyisipkan data kedalam nilai prediksi perubahan piksel (*expanded prediction error*). PEE kemudian dikembangkan lebih lanjut dalam hal pemilihan piksel referensi dan penentuan tipe blok pada *carrier*. Namun pengembangan yang telah diusulkan belum memanfaatkan blok secara maksimal. Dalam satu blok, jumlah pesan yang mampu disisipkan hanya berkisar antara 2-6 bit saja. Salah satu cara yang dapat digunakan untuk mengatasi masalah tersebut adalah mengelompokkan piksel bernilai mirip secara adaptif dan menyisipkan 2 bit data sekaligus dalam sebuah piksel tanpa menggunakan konsep *multi-layer embedding*. Untuk menjaga agar perubahan nilai tidak terlalu besar, diusulkan metode reduksi pada nilai *expanded prediction error*. Berdasarkan hasil pengujian, metode yang diusulkan mampu meningkatkan kualitas dan kapasitas citra stego dengan rata-rata sebesar 4,257 dB dan 156.776 bit.

**Kata Kunci:** *Data Hiding, Pengelompokan Nilai Piksel, Prediction Error Expansion, Reduksi Error Expansion.*

*[Halaman ini sengaja dikosongkan]*

# **Peningkatan Kinerja *Prediction Error Expansion* dalam Data Hiding dengan Mereduksi *Error Expansion* dan Mengelompokkan Nilai Piksel Secara Adaptif**

Nama Mahasiswa : Hendro Eko Prabowo  
NRP : 5116 201 006  
Pembimbing : Tohari Ahmad, S.Kom., M.IT., Ph.D.

## **ABSTRACT**

Today, the exchange of information is easier because of the rapid development of information and communication technology. This condition is proven by the increasing of the use of global IP of 88,7 EB (exabytes) per month in 2016. This increasement makes information security be the one of most important requirement in securing sensitive data such as corporate financial data or national security data. The security of sensitive data can be achieved by using a technique called data hiding. One of data hiding technique is steganography that embeds data using mathematical algorithms into multimedia object (e.g. digital image, video, audio) called *carrier*. The mathematical algorithm modifies the pixel values in the digital image. The modification should not change the carrier image significantly and provide large embedding capacity. This develop requirement makes algorithms that used in data hiding technique is developed continuously.

One of the mathematical algorithms for data hiding is *prediction error expansion (PEE)*. It embeds data into predicted pixel value (*expanded prediction error*). PEE was developed further in terms of reference pixel selection and block type determination on *carrier* images. However, the existing PEE algorithm has no use the block maximally. In a block, the existing PEE algorithm only can embed 2-6 bits of data. Therefore, it is necessary to develop the utilization of blocks. One of methods that can be used is grouping similar pixel value adaptively and embedding 2-bits of data into a block without using multi-layer embedding concept. Additionally, to keep the pixel value from being changed significantly while hiding data, the expanded prediction error value will be reduced. Based on experiment, it shows that the proposed method can improve quality and capacity of stego image by an average of 4,257 dB and 156.776 bit.

**Keywords:** *Data Hiding, Pixel Value Grouping, Prediction Error Expansion, Reduction Error Expansion.*

*[Halaman ini sengaja dikosongkan]*



## KATA PENGANTAR

Alhamdulillahirabbil'alamin. Segala puji bagi Allah SWT, Tuhan semesta alam yang telah melimpahkan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan Tesis yang berjudul “Peningkatan Kinerja *Prediction Error Expansion* dalam *Data Hiding* dengan Mereduksi *Error Expansion* dan Mengelompokkan Nilai Piksel Secara Adaptif” dengan hasil yang memuaskan.

Pengerjaan Tesis ini merupakan kesempatan yang sangat berharga bagi penulis karena dalam proses pengerjaan Tesis, penulis dapat memahami, meningkatkan serta menerapkan ilmu yang telah didapatkan selama menempuh pendidikan di Program Pasca Sarjana Teknik Informatika ITS. Penulis menyadari bahwa penulisan ini tidak akan terwujud tanpa adanya bantuan dan dorongan dari berbagai pihak. Oleh karena itu penulis menyampaikan terima kasih kepada :

1. Ibu Siti Rupi'ah dan Bapak Tarmani selaku orang tua tercinta, atas segala yang telah diberikan dan tak terhitung banyaknya dan yang selalu memberikan do'a serta bimbingannya sampai ananda berhasil,
2. Guntur Yudha Mahendra selaku adik penulis yang selalu mendukung dan memberikan semangat serta do'a.
3. Siti Zumrotul Ainiyah, teman terdekat yang menjadi teman diskusi dan pemberi semangat.
4. Bapak Tohari Ahmad, S.Kom., MIT., Ph.D, selaku pembimbing yang telah memberikan kepercayaan, motivasi, bimbingan, nasihat dan bantuan kepada penulis dalam menyelesaikan Tesis ini.
5. Bapak Waskitho Wibisono, S.Kom., M.Eng., Ph.D. selaku dosen penguji dan Ketua Program Pasca Sarjana Teknik Informatika ITS.
6. Bapak Prof. Ir. Supeno Djanali, M.Sc. Ph.D dan Bapak Dr.Eng. Radityo Anggoro, S.Kom., M.Sc. selaku dosen penguji yang telah memberikan saran, arahan dan koreksi dalam pengerjaan Tesis ini.

7. Mbak Lina, Mas Kunto dan segenap Staff Tata Usaha yang telah memberikan bantuan dan kemudahan penulis selama menjalani kegiatan belajar di Teknik Informatika ITS.
8. Rekan-rekan Teknik Informatika angkatan 2012-2016 baik dari Program Pasca Sarjana maupun program Sarjana yang telah membantu penulis dalam proses penyelesaian Tesis dan sedang berjuang bersama untuk mendapatkan kesuksesan.

Penulis menyadari bahwa Buku Tesis ini masih jauh dari kata sempurna. Seluruh kritik dan saran dari pembaca akan diterima dengan pikiran terbuka oleh penulis. Semoga buku ini bermanfaat dan memberikan kontribusi yang berarti bagi pihak yang membutuhkan.

Surabaya, 1 Januari 2018

Penulis

## DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	iii
ABSTRAK.....	v
ABSTRACT.....	vii
KATA PENGANTAR .....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL.....	xv
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah.....	4
1.3 Tujuan Penelitian.....	4
1.4 Manfaat Penelitian.....	5
1.5 Kontribusi Penelitian.....	5
1.6 Batasan Masalah.....	5
BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI.....	7
2.1 Dasar Teori .....	7
2.1.1 Steganografi .....	7
2.1.2 Citra Skala Keabuan ( <i>Grayscale Image</i> ).....	8
2.1.3 <i>Peak Signal to Noise Ratio (PSNR)</i> .....	9
2.2 Studi Literatur .....	9
2.2.1 <i>Prediction-Error Expansion (PEE)</i> .....	9
2.2.2 <i>Pixel Value Ordering (PVO) based PEE Embedding</i> .....	11
2.2.3 <i>Improving Pixel Value Ordering (PVO) based PEE Embedding</i> .....	13
2.2.4 <i>Adaptive Pixel-Embedding Strategy dalam PVO-based PEE Embedding</i> .....	15
2.2.5 <i>Pixel Value Grouping (PVG)</i> .....	17
BAB 3 METODE PENELITIAN.....	19

3.1 Tahapan Penelitian.....	19
3.2 Hasil Kajian Tentang <i>Prediction-Error Expansion</i> .....	19
3.3 Perancangan Algoritma Penyisipan .....	21
3.3.1 Pembagian Citra <i>Carrier</i> .....	21
3.3.2 Penyisipan Data .....	24
3.3.3 Pembuatan Citra <i>Stego</i> .....	28
3.3.4 <i>Location Map (LM)</i> .....	28
3.4 Perancangan Algoritma Ekstraksi.....	30
3.4.1 Pembagian Citra <i>Stego</i> .....	30
3.4.2 Pengembalian Nilai Piksel dan Ekstraksi Data.....	32
3.4.3 Pembangunan Citra <i>Carrier</i> .....	34
3.5 Rencana Implementasi Algoritma .....	35
3.6 Pengujian .....	35
3.7 Analisa Hasil.....	36
BAB 4 HASIL DAN PEMBAHASAN .....	37
4.1 Lingkungan Uji Coba dan Implementasi Algoritma .....	37
4.1.1 Lingkungan Uji Coba .....	37
4.1.2 Implementasi <i>Adaptive Pixel Value Grouping</i> .....	37
4.1.3 Implementasi Reduksi <i>Error Expansion</i> .....	38
4.2 Kualitas dan Kapasitas Citra <i>Stego</i> .....	40
4.3 Pengaruh Perubahan Ukuran Blok dan <i>Embedding Level</i> .....	41
4.3.1 Perubahan Ukuran Blok.....	42
4.3.2 Perubahan <i>Embedding Level</i> .....	44
BAB 5 KESIMPULAN DAN SARAN .....	47
5.1 Kesimpulan .....	47
5.2 Saran .....	47
DAFTAR PUSTAKA.....	49
LAMPIRAN .....	53
BIODATA PENULIS .....	56

## DAFTAR GAMBAR

Gambar 2.1 Penggambaran Steganografi dalam <i>The Prisoner's Problem</i> .....	7
Gambar 2.2 Skema Umum Steganografi .....	8
Gambar 2.3 Contoh Citra Skala Keabuan .....	9
Gambar 2.4 Blok Piksel dalam Konsep PEE .....	10
Gambar 2.5 Blok Piksel Berukuran $r$ Baris dan $c$ Kolom .....	15
Gambar 2.6 <i>Pixel Value Grouping (PVG)</i> .....	17
Gambar 3.1 Diagram Alur Penelitian.....	19
Gambar 3.2 Alur Perancangan dan Penerapan Algoritma Penyisipan Data .....	20
Gambar 3.3 Diagram Alur Pembentukan Kelompok Piksel .....	22
Gambar 3.4 Proses Penyisipan Data .....	24
Gambar 3.5 Piksel Prediktor pada Tipe <i>Smooth</i> .....	26
Gambar 3.6 Alur Ekstraksi Data dan Pengembalian Nilai Piksel.....	30
Gambar 3.7 Diagram Alur Pembentukan Kelompok Piksel <i>PG</i> .....	31
Gambar 3.8 Citra <i>Carrier</i> .....	34
Gambar 4.1 <i>Pseudo-code</i> Algoritma <i>Adaptive Pixel Value Grouping (APVG)</i> ....	38
Gambar 4.2 <i>Pseudo-code</i> Algoritma Reduksi <i>Error Expansion</i> .....	39
Gambar 4.3 Pengaruh Perubahan Blok Terhadap Kualitas Citra Stego.....	43
Gambar 4.4 Pengaruh <i>Embedding Level</i> Terhadap Kualitas Citra Stego .....	45

*[Halaman ini sengaja dikosongkan]*

## DAFTAR TABEL

Tabel 3.1 Penggunaan Location Map .....	29
Tabel 3.2 Skema Pengujian.....	36
Tabel 4.1 Rincian Lingkungan Uji Coba .....	37
Tabel 4.2 Perbandingan Kualitas Citra Stego .....	40
Tabel 4.3 Perbandingan Kapasitas .....	41
Tabel 4.4 Kualitas Citra Stego ‘Lena’ Terhadap Perubahan Blok.....	44
Tabel 4.5 Kualitas Citra Stego dan Jumlah Shifted Pixel pada Beberapa Carrier	46

*[Halaman ini sengaja dikosongkan]*



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi informasi memberikan dampak positif dalam kehidupan manusia, salah satunya adalah pertukaran informasi menjadi lebih mudah. Hal ini dapat dilihat dari meningkatnya penggunaan internet dari tahun ke tahun. Menurut Cisco Visual Network Index (Cisco VNI), trafik *IP global* meningkat sebesar 88,7 *exabytes* (EB) perbulan pada tahun 2016 dan diperkirakan pada tahun 2020 akan mencapai 194 EB perbulan (Cisco, 2017). Semakin meningkatnya pengguna internet untuk bertukar informasi, keamanan informasi menjadi salah satu aspek terpenting terutama untuk data-data sensitif seperti data finansial perusahaan dan data yang berhubungan dengan keamanan negara. Salah satu cara yang dapat digunakan untuk mengamankan data saat bertukar melalui internet adalah menyembunyikan data tersebut kedalam objek digital.

Teknik yang dapat digunakan untuk mengamankan informasi tersebut adalah *data hiding*. Metode ini akan menyembunyikan informasi (data) dalam sebuah objek (*carrier*) seperti citra digital, melibatkan algoritma matematis untuk memodifikasi piksel dan menghasilkan objek yang disebut objek *stego* (*stego object*) (Wu et al., 2003; Ni et al., 2006; Mukhopadhyay, 2011; Rad et al., 2016). Modifikasi yang dilakukan sebaiknya menjaga agar *carrier* tidak berubah atau mendapatkan kerusakan yang cukup signifikan. Disisi lain, algoritma *data hiding* harus mampu menyediakan kapasitas penyimpanan informasi atau data yang besar (Wu, 2002; Arya et al., 2016; Andra et al., 2017). Kondisi ini yang membuat metode *data hiding* terus dikembangkan dengan capaian agar mendapatkan kapasitas penyimpanan pesan yang besar dengan tingkat kerusakan citra *stego* yang rendah.

Banyak sekali metode *data hiding* yang telah dikembangkan salah satunya adalah *Different Expansion (DE)*. DE pertama kali diusulkan oleh Tian dengan mengambil kelebihan dari nilai perbedaan antara dua piksel yang bertetangga (Tian, 2003). Nilai tersebut kemudian digunakan untuk menyimpan pesan dan

ditambahkan dengan dalam piksel lama untuk menghasilkan piksel baru. Metode ini dapat menyisipkan pesan pada *carrier* yang digunakan dengan rata-rata 0.5 *bit per pixel (bpp)*. Karena rendahnya kapasitas penyimpanan yang didapatkan, penelitian lain meningkatkan DE dalam berbagai aspek seperti *integer transform generalization* (Gui et al., 2012; Peng et al., 2012), *location map reduction* (Kim et al., 2008; Liu et al., 2012), *prediction error expansion (PEE)* (Thodi & Rodriguez, 2007; Sachnev et al., 2009; Dragoi & Coltuc, 2014), *pixel selection embedding* (Li et al., 2011), dan *pixel value ordering* (Li et al., 2013; Wang et al., 2015).

Beberapa pengembangan diatas, salah satu yang menarik perhatian adalah *prediction error expansion (PEE)*. PEE memanfaatkan nilai prediksi perubahan piksel (*expanded prediction error*) untuk menyisipkan data sebesar 1 bit. Sehingga perubahan nilai piksel tidak terlalu besar dan dapat menjaga tingkat kerusakan piksel tetap rendah. Namun ada beberapa kekurangan dari PEE diantaranya adalah kapasitas pesan yang dapat disisipkan masih terlalu rendah yaitu 0.561 bit per pixel (bpp). Hal ini diperburuk dengan tingkat kerusakan dari citra *stego* sekitar 46 dB yang diukur menggunakan metode *pixel noise to signal ratio (PSNR)* (Thodi & Rodriguez, 2007).

Keadaan tersebut kemudian dikembangkan lebih lanjut dengan cara membagi histogram dari nilai *expanded prediction error* menjadi dua kelompok yaitu *inner region* dan *outer region*. Piksel dalam *inner region* digunakan untuk penyisipan data, sedangkan *outer region* akan dilakukan pergeseran nilai piksel atau *shifting*. Pembagian ini bertujuan agar jumlah piksel untuk penyisipan data semakin meningkat. Hasil yang didapatkan cukup memuaskan dengan meningkatnya kapasitas penyisipan data dengan tingkat kerusakan citra *stego* yang sama (Hu et al., 2009).

Pengembangan metode PEE yang lebih baik diusulkan oleh Li, et al (2013) dengan melakukan pemilihan piksel prediktor untuk menghitung nilai *expanded prediction error*. Langkah yang dilakukan adalah membagi citra *carrier* menjadi blok-blok tidak berpotongan dan masing-masing blok tersebut diurutkan (*sort*) dari nilai kecil ke nilai besar (*ascending*). Piksel prediktor yang dihasilkan berjumlah

dua jenis, yaitu piksel prediktor minimum dan maksimum. Piksel prediktor minimum untuk menghitung nilai *expanded prediction error* dari piksel bernilai kecil sedangkan piksel prediktor maksimum untuk menghitung dari piksel bernilai besar. Hasilnya adalah metode ini mampu meningkatkan kualitas citra *stego* sekitar 2-3 dB lebih baik dengan metode pengukuran yang sama.

Metode Li, et al (2013) kemudian dilakukan penelitian lebih lanjut untuk meningkatkan kapasitas penyisipan data dengan cara mengusulkan metode baru untuk menghitung *expanded prediction error* (Peng et al., 2014). Selain hal tersebut, mereka juga mengusulkan cara pemilihan blok yang halus (*smooth*) untuk penyisipan data. Sedangkan blok yang tidak halus tidak digunakan untuk penyisipan. Kedua hal ini berhasil meningkatkan kualitas citra *stego* dengan data yang disisipkan berjumlah sama.

Usulan model penentuan blok *smooth* yang lain adalah menggunakan skema hubungan piksel bertetangga dari sebuah blok piksel (Weng et al., 2016). Hasil dari skema yang diusulkan adalah blok dengan tipe *smooth* dan *complex*. Blok tipe *smooth* dibagi lebih lanjut menjadi beberapa tingkatan berdasarkan korelasi dengan piksel tetangganya. Tingkatan tersebut adalah *high correlation*, *moderate correlation* dan *low correlation*. Jumlah data yang disisipkan juga dibedakan untuk masing-masing tingkat korelasi. *High correlation* mendapatkan porsi yang paling besar sedangkan *low correlation* mendapatkan porsi yang paling kecil. Tindakan ini dilakukan demi menjaga agar kualitas citra *stego* semakin baik.

Namun dari berbagai usulan pengembangan metode PEE masih belum memanfaatkan blok secara maksimal. Li, et al (2013) dan Peng, et al (2014) hanya menggunakan nilai piksel terbesar dan piksel terkecil untuk menyisipkan data. Contohnya adalah jika terdapat blok berukuran 3x3, maka hanya digunakan 2 piksel dari 9 piksel dalam blok untuk penyisipan data. Hal ini berarti, satu blok maksimal menyisipkan hanya 2 bit data. Sedangkan metode Weng, et al (2016), menggunakan paling banyak 3 piksel dari kelompok minimum dan maksimum. Sehingga metode tersebut hanya mampu menyisipkan maksimal 6 bit dalam satu blok. Uraian diatas menunjukkan bahwa penggunaan blok untuk penyisipan data belum maksimal.

Oleh karena itu, perlu diketahui metode baru untuk memaksimalkan penggunaan blok dalam penyisipan data.

Salah satu cara yang dapat digunakan untuk memaksimalkan penggunaan blok adalah dengan mengelompokkan piksel-piksel yang bernilai mirip secara adaptif menjadi satu kelompok. Pengelompokkan yang bersifat adaptif ini dapat membentuk kelompok dari sebuah blok dengan jumlah kelompok yang berbeda-beda berdasarkan kondisi blok tersebut. Hal lain yang dapat dilakukan adalah mengembangkan teknik baru untuk menyisipkan data lebih dari 1 bit misal 2 bit sekaligus dalam sekali proses penyisipan. Perihal penyisipan data sebesar 2 bit dalam sekali proses penyisipan akan berefek pada nilai *expanded prediction error* yang semakin besar dan tingkat kerusakan citra *stego* semakin tinggi. Kondisi ini dapat dikurangi dengan mereduksi nilai *expanded prediction error* untuk menjaga tingkat kerusakan citra *stego* tersebut.

## **1.2 Perumusan Masalah**

Rumusan masalah yang diangkat dalam penelitian ini adalah sebagai berikut :

1. Bagaimana mengelompokkan piksel yang bernilai mirip secara adaptif menjadi satu kelompok ?
2. Bagaimana pengaruh reduksi nilai *expanded prediction error* terhadap tingkat kerusakan citra *stego* yang dihasilkan ?
3. Bagaimana pengaruh menggunakan metode pengelompokkan piksel yang bersifat adaptif terhadap kapasitas citra *stego* ?
4. Bagaimana pengaruh besar blok terhadap tingkat kerusakan citra *stego* ?

## **1.3 Tujuan Penelitian**

Tujuan yang ingin dicapai dalam pembuatan penelitian ini adalah untuk menghasilkan metode baru dalam *data hiding* yang dapat mengelompokkan piksel bernilai mirip secara adaptif dan dapat menyisipkan data sebesar 2 bit pada sebuah

piksel. Berkaitan dengan penyisipan data tersebut, diharapkan metode baru yang dimaksud dapat mereduksi nilai *expanded prediction error* yang semakin besar.

#### **1.4 Manfaat Penelitian**

Manfaat dari penelitian ini adalah memberikan metode *data hiding* yang menghasilkan kualitas citra *stego* lebih baik dengan ditandai meningkatnya nilai hasil pengukuran menggunakan metode *pixel signal to noise ratio (PSNR)*.

#### **1.5 Kontribusi Penelitian**

Kondisi terkini adalah metode *data hiding* yang diusulkan sebelumnya masih belum maksimal dalam memanfaatkan penggunaan blok sehingga tingkat kerusakan objek *stego* masih cukup tinggi. Penelitian ini mengembangkan metode baru untuk mengurangi permasalahan tersebut. Kontribusi dari penelitian ini adalah mengembangkan metode baru untuk mengelompokkan piksel bernilai mirip secara adaptif serta mereduksi nilai *expanded prediction error* setelah dilakukan penyisipan data sebesar 2 bit.

#### **1.6 Batasan Masalah**

Batasan masalah pada penelitian ini adalah :

1. Objek *carrier* yang digunakan adalah citra digital berskala keabuan.
2. Citra digital tersebut berukuran 512x512 piksel yang diambil dari USC-SIPI (USC-SIPI, 1977) dan eMicrobes Digital Library (System Partners Healthcare Inc., 2002). Penggunaan citra dari USC-SIPI mengikuti Peng et al (2014) yang menjadi salah satu referensi dalam penelitian.
3. Pesan yang digunakan berupa pesan bernilai biner dan teks dengan ukuran yang telah ditentukan.
4. Metode pengembangan diimplementasikan menggunakan Matlab versi 2017a.

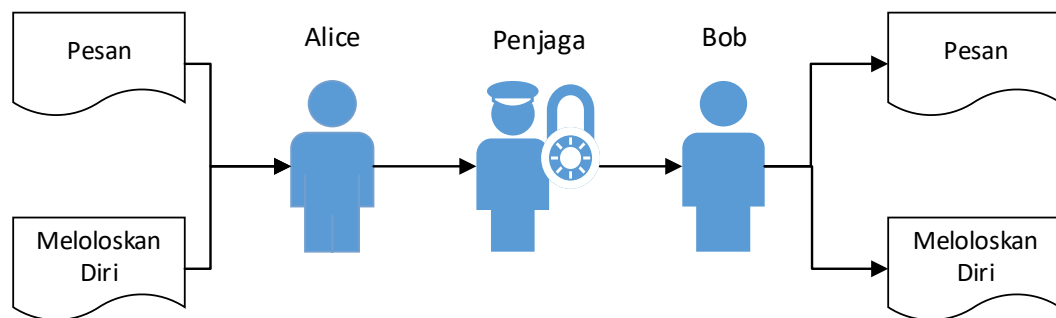
*[Halaman ini sengaja dikosongkan]*

## BAB 2

## 2.1 Dasar Teori

### 2.1.1 Steganografi

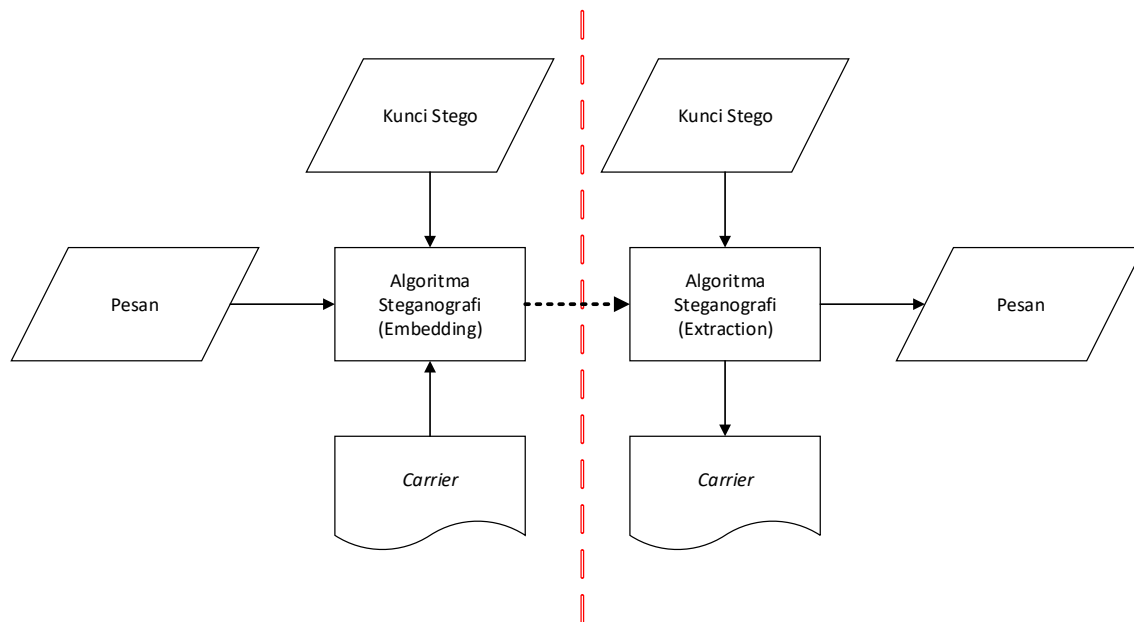
Steganografi atau *data hiding* merupakan metode untuk menyembunyikan data dengan cara menyisipkannya ke dalam medium tertentu dan mentransmisikannya. Tujuan penyisipan data ini agar keberadaan data dalam medium tidak diketahui (Al-Ani et al., 2010). Konsep dasar steganografi merujuk pada penelitian yang memperkenalkan konsep '*Prisoner's Problem*' (Simmons, 1984). Konsep ini memisalkan steganografi dalam bentuk komunikasi dalam bentuk pesan antara dua tahanan yang bernama Alice dan Bob yang ingin meloloskan diri. Namun komunikasi mereka selalu diperhatikan oleh penjaga. Sehingga Alice dan Bob perlu menyembunyikan pesan perihal meloloskan diri kedalam pesan yang tidak mencurigakan. Konsep inilah yang dinamakan steganografi.



Gambar 2.1 Penggambaran Steganografi dalam *The Prisoner's Problem*

Dalam steganografi, informasi atau data yang ingin disisipkan memiliki batasan berdasarkan besarnya medium atau *carrier* yang digunakan (Artz, 2001). Semakin besar *carrier* yang digunakan maka semakin besar pula peluang menyisipkan pesan dalam jumlah besar. Namun, dalam proses penyisipan juga perlu diperhatikan agar *carrier* tidak terdegradasi atau mengalami kerusakan

sehingga bentuk *carrier* menjadi mencurigakan (Bender et al., 1996). Skema umum untuk proses penyisipan pesan dalam steganografi dapat dilihat pada Gambar 2.2. Selain terdapat proses penyisipan pesan, Gambar 2.2 juga menunjukkan proses ekstraksi dari untuk mendapatkan pesan atau data yang disisipkan dalam citra. Sehingga dalam steganografi memiliki dua proses utama yaitu, penyisipan pesan (*embedding*) dan ekstraksi pesan (*extraction*).



Gambar 2.2 Skema Umum Steganografi

### 2.1.2 Citra Skala Keabuan (*Grayscale Image*)

Citra skala keabuan atau *grayscale image* adalah citra yang piksel-pikselnya hanya memiliki satu intensitas warna. Intensitas warna terkuat adalah putih sedangkan intensitas warna terlemah adalah hitam. Sehingga citra ini hanya terdiri dari dua warna yaitu hitam atau putih dengan intensitas yang berbeda-beda. Jika direpresentasikan dalam nilai 8-bit, warna hitam memiliki nilai 0 sedangkan warna putih memiliki nilai 255. Contoh citra skala keabuan dapat dilihat pada Gambar 2.3.





Gambar 2.3 Contoh Citra Skala Keabuan

### 2.1.3 Peak Signal to Noise Ratio (PSNR)

*Peak Signal to Noise Ratio (PSNR)* merupakan salah satu model pengukuran untuk mengukur tingkat derau (*noise*) yang terdapat dalam sebuah objek. Derau ini ada karena terdapat perubahan nilai pada sebuah objek akibat proses tertentu misalnya perubahan nilai piksel pada citra efek dari penyisipan data. Pengukuran menggunakan PSNR juga sering digunakan untuk mengukur kualitas dari citra *stego* dalam bidang steganografi. Semakin tinggi nilai PSNR menandakan citra *stego* semakin mirip dengan citra asli (Al-Najjar & Soong, 2012; Lahoulu et al., 2013). Tingkat kemiripan ini yang mencerminkan kualitas dari citra *stego*. PSNR dapat dihitung menggunakan persamaan berikut :

$$PSNR = 10 \log_{10} \frac{Max(X)^2}{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.1)$$

dengan  $Max(X)$  adalah nilai intensitas maksimum dari sebuah citra,  $n$  adalah jumlah piksel dari citra yang diukur,  $x_i$  adalah piksel dari citra asli dan  $y_i$  adalah piksel dari citra *stego*.

## 2.2 Studi Literatur

### 2.2.1 Prediction-Error Expansion (PEE)

*Prediction-Error Expansion* pertama kali diusulkan oleh Thodi dan Rodriguez (2007) yang memanfaatkan piksel bertetangga untuk dihitung nilai prediksi perubahan piksel atau *expanded prediction error (p)*. Nilai  $p$  tersebut kemudian digunakan untuk menyisipkan 1 bit informasi atau data. Proses

penghitungan nilai  $p$  dilakukan dengan menggunakan persamaan (2.2) dan melibatkan 4 piksel yang dapat dilihat pada Gambar 2.4.

$c1$	$c2$
$c3$	$a$

Gambar 2.4 Blok Piksel dalam Konsep PEE

$$p = a - \hat{a} \quad (2.2)$$

Dengan :

$a$  = piksel yang digunakan untuk menyisipkan data

$\hat{a}$  = nilai prediktor

$\hat{a}$  didapatkan dengan menggunakan persamaan dibawah ini:

$$\tilde{a} = \begin{cases} \max(c2, c3), & \text{jika } c1 \leq \min(c2, c3) \\ \min(c2, c3), & \text{jika } c1 \geq \max(c1, c3) \\ c2 + c3 - c1, & \text{sebaliknya} \end{cases} \quad (2.3)$$

$$\hat{a} = 2[\tilde{a}/2] \quad (2.4)$$

dengan  $c1, c2, c3$  adalah piksel-piksel referensi dari blok PEE yang diperlihatkan pada Gambar 2.4. Proses penyisipan informasi ( $m$ ) dapat diselesaikan dengan menggunakan persamaan (2.5) dan nilai piksel baru ( $a'$ ) didapatkan dengan persamaan (2.6).

$$p' = 2p + m \quad (2.5)$$

$$a' = a + p' \quad (2.6)$$

dengan  $p'$  adalah nilai *expanded prediction error* yang baru setelah disisipi pesan.

Konsep PEE kemudian dikembangkan lebih lanjut dengan membagi histogram dari nilai  $p$  menjadi dua bagian, yaitu *inner region* yang digunakan untuk penyisipan data dan *outer region* sebagai daerah yang akan digeser (*shifted*) nilainya. Pembagian ini dilakukan dengan menyertakan nilai  $T_l$  sebagai pembatas kiri dan  $T_r$  sebagai pembatas kanan dari *inner region*. Selain membagi daerah histogram dari  $p$ , pengembangan juga dilakukan dengan mengembangkan persamaan (2.5) menjadi persamaan (2.7) dan untuk mendapatkan nilai piksel baru pada *inner region* dapat digunakan persamaan (2.8) (Hu, et al., 2009).

$$p' = p + m \quad (2.7)$$

$$a' = a + p', \quad \text{jika } -T_l \leq p' \leq T_r - 1 \quad (2.8)$$

Sedangkan untuk *outer region*, akan dilakukan penggeseran nilai (*shifting*) dengan menggunakan persamaan (2.9).

$$a' = \begin{cases} x + T_r, & \text{jika } p_{mr} \geq p' \geq T_r \\ x - T_l, & \text{jika } -p_{ml} \leq p' \leq -T_l - 1 \end{cases} \quad (2.9)$$

dengan  $p_{mr}$  adalah nilai akhir dari *outer region* sebelah kanan dan  $p_{ml}$  adalah nilai akhir dari *outer region* sebelah kiri.

### 2.2.2 Pixel Value Ordering (PVO) based PEE Embedding

*Pixel Value Ordering (PVO)* merupakan salah satu pengembangan dari PEE yang memanfaatkan konsep pengurutan nilai piksel dari kecil ke besar (*ascending sort*). Sebelumnya, citra yang digunakan sebagai *carrier* dibagi menjadi blok-blok tidak saling beririsan dengan ukuran yang sama. Masing-masing blok tersebut kemudian di-*sorting* secara *ascending* sehingga menghasilkan urutan piksel ( $P_{\sigma(1)}, \dots, P_{\sigma(n)}$ ) dengan  $n$  adalah jumlah piksel dalam satu blok (Li, et al., 2013). Nilai *expanded prediction error (PE)* pada pengembangan ini dibedakan menjadi dua yaitu  $PE_{max}$  dan  $PE_{min}$ . Masing-masing nilai  $PE$  akan digunakan

untuk menyisipkan data kedalam piksel dengan nilai terbesar dan piksel dengan nilai terkecil. Persamaan (2.10)-(2.11) dapat digunakan untuk mendapatkan kedua nilai  $PE$  tersebut.

$$PE_{max} = x_{\sigma(n)} - x_{\sigma(n-1)} \quad (2.10)$$

$$PE_{min} = x_{\sigma(1)} - x_{\sigma(2)} \quad (2.11)$$

dengan  $x_{\sigma(n)}$  adalah nilai piksel terbesar,  $x_{\sigma(n-1)}$  adalah nilai piksel terbesar kedua,  $x_{\sigma(1)}$  adalah nilai terkecil dan  $x_{\sigma(2)}$  adalah nilai terkecil kedua. Sedangkan untuk proses penyisipan data dan penghitungan nilai piksel setelah disisipi dapat menggunakan persamaan (2.12)-(2.15). Dengan menerapkan pengembangan tersebut, Li, et al. (2013) mampu mengurangi jumlah piksel yang harus digeser (*shifted*) dan menurut mereka hal tersebut berarti semakin sedikit piksel yang harus dimodifikasi pada *carrier*. Sehingga kemampuan dari algoritma yang diusulkan akan semakin baik berkaitan dengan tingkat kerusakan yang dihasilkan.

$$\widetilde{PE}_{max} = \begin{cases} PE_{max}, & \text{jika } PE_{max} = 0 \\ PE_{max} + b, & \text{jika } PE_{max} = 1 \\ PE_{max} + 1, & \text{jika } PE_{max} > 1 \end{cases} \quad (2.12)$$

$$\widetilde{PE}_{min} = \begin{cases} PE_{min}, & \text{jika } PE_{min} = 0 \\ PE_{min} - b, & \text{jika } PE_{min} = -1 \\ PE_{min} - 1, & \text{jika } PE_{min} > -1 \end{cases} \quad (2.13)$$

$$\tilde{x} = x_{\sigma(n-1)} + \widetilde{PE}_{max} = \begin{cases} x_{\sigma(n)}, & \text{jika } PE_{max} = 0 \\ x_{\sigma(n)} + b, & \text{jika } PE_{max} = 1 \\ x_{\sigma(n)} + 1, & \text{jika } PE_{max} > 1 \end{cases} \quad (2.14)$$

$$\tilde{x} = x_{\sigma(2)} - \widetilde{PE}_{min} = \begin{cases} x_{\sigma(1)}, & \text{jika } PE_{min} = 0 \\ x_{\sigma(1)} - b, & \text{jika } PE_{min} = -1 \\ x_{\sigma(1)} - 1, & \text{jika } PE_{min} > -1 \end{cases} \quad (2.15)$$

### 2.2.3 Improving Pixel Value Ordering (PVO) based PEE Embedding

Pengembangan PVO-based ini memanfaatkan *expanded prediction error* yang bernilai 0 untuk penyisipan data (Peng et al., 2014). Hal ini mengakibatkan dengan jumlah data yang disisipkan sama, metode yang diajukan Peng, et al memiliki tingkat kerusakan yang lebih kecil. Kondisi ini dibuktikan dengan nilai PSNR yang lebih besar dibandingkan dengan PVO-based dari Li, et al. (2013). Nilai *expanded prediction error* hasil dari pengembangan ini dihitung dengan menggunakan persamaan (2.16)-(2.17).

$$d_{max} = x_u - x_v, \text{ dengan } \begin{cases} u = \min(\sigma(n), \sigma(n-1)) \\ v = \max(\sigma(n), \sigma(n-1)) \end{cases} \quad (2.16)$$

$$d_{min} = x_s - x_t, \text{ dengan } \begin{cases} s = \min(\sigma(1), \sigma(2)) \\ t = \max(\sigma(1), \sigma(2)) \end{cases} \quad (2.17)$$

dengan  $x$  adalah piksel pada blok setelah dilakukan proses *sorting* sedangkan  $\sigma(n)$ ,  $\sigma(n-1)$ ,  $\sigma(1)$  dan  $\sigma(2)$ , masing-masing adalah indeks piksel terbesar, terbesar kedua, terkecil dan terkecil kedua dalam urutan piksel sebelum di-*sorting*.

Hasil pengembangan dari  $d_{max}$  dan  $d_{min}$  juga berpengaruh terhadap proses penyisipan data dan penghitungan nilai baru. Masing-masing proses tersebut dapat diselesaikan dengan persamaan (2.18)-(2.19) dan persamaan (2.20)-(2.21).

$$\tilde{d}_{max} = \begin{cases} d_{max} + b, & \text{jika } d_{max} = 1, \\ d_{max} + 1, & \text{jika } d_{max} > 1, \\ d_{max} - b, & \text{jika } d_{max} = 0, \\ d_{max} - 1, & \text{jika } d_{max} < 0. \end{cases} \quad (2.18)$$

$$\tilde{d}_{min} = \begin{cases} d_{min} + b, & \text{jika } d_{min} = 1, \\ d_{min} + 1, & \text{jika } d_{min} > 1, \\ d_{min} - b, & \text{jika } d_{min} = 0, \\ d_{min} - 1, & \text{jika } d_{min} < 0. \end{cases} \quad (2.19)$$

$$\tilde{x} = x_{\sigma(n-1)} + |\tilde{d}_{max}| = \begin{cases} x_{\sigma(n)} + b, & \text{jika } d_{max} = 1, \\ x_{\sigma(n)} + 1, & \text{jika } d_{max} > 1, \\ x_{\sigma(n)} + b, & \text{jika } d_{max} = 0, \\ x_{\sigma(n)} + 1, & \text{jika } d_{max} < 0. \end{cases} \quad (2.20)$$

$$\tilde{x} = x_{\sigma(2)} - |\tilde{d}_{min}| = \begin{cases} x_{\sigma(1)} - b, & \text{jika } d_{min} = 1, \\ x_{\sigma(1)} - 1, & \text{jika } d_{min} > 1, \\ x_{\sigma(1)} - b, & \text{jika } d_{min} = 0, \\ x_{\sigma(1)} - 1, & \text{jika } d_{min} < 0. \end{cases} \quad (2.21)$$

Sedangkan untuk proses ekstraksi pesan dan pengembalian nilai piksel, metode ini memerlukan nilai selisih piksel referensi dan piksel untuk penyisipan. Nilai tersebut didapatkan dari persamaan berikut :

$$\tilde{d}_{max} = y_u - y_v \quad (2.22)$$

$$\tilde{d}_{min} = y_s - y_t \quad (2.23)$$

Dengan :

$y$  = piksel dari citra *stego*

$u, v, s, t$  = variabel yang didefinisikan pada persamaan (2.16)-(2.17)

Jika telah didapatkan kedua variabel diatas, maka untuk mendapatkan pesan yang disisipkan dan nilai dari piksel asli dapat menggunakan ketentuan dibawah ini:

a. Untuk  $\tilde{d}_{max}$  :

- Jika  $\tilde{d}_{max} \in \{1,2\}$  maka data yang disisipkan adalah  $b = \tilde{d}_{max} - 1$  dan nilai piksel asli adalah  $x_{\sigma(n)} = y_u - b$ .
- Jika  $\tilde{d}_{max} > 2$ , maka tidak ada data yang disisipkan dan nilai piksel asli adalah  $x_{\sigma(n)} = y_u - 1$ .
- Jika  $\tilde{d}_{max} \in \{0, -1\}$  maka data yang disisipkan adalah  $b = -\tilde{d}_{max}$  dan nilai piksel asli adalah  $x_{\sigma(n)} = y_v - b$ .

- Jika  $\tilde{d}_{max} < -1$ , maka tidak ada data yang disisipkan dan nilai piksel asli adalah  $x_{\sigma(n)} = y_v - 1$ .
- b. Untuk  $\tilde{d}_{min}$  :
  - Jika  $\tilde{d}_{min} \in \{1,2\}$ , maka data yang disisipkan adalah  $b = \tilde{d}_{min} - 1$  dan nilai piksel asli adalah  $x_{\sigma(1)} = y_t + b$ .
  - Jika  $\tilde{d}_{min} > 2$ , maka data tidak ada data yang disisipkan dan nilai piksel asli adalah  $x_{\sigma(1)} = y_t + 1$ .
  - Jika  $\tilde{d}_{min} \in \{0, -1\}$ , maka data yang disisipkan adalah  $b = -\tilde{d}_{min}$  dan nilai piksel asli adalah  $x_{\sigma(1)} = y_s + b$ .
  - Jika  $\tilde{d}_{min} < -1$ , maka data tidak ada data yang disisipkan dan nilai piksel asli adalah  $x_{\sigma(1)} = y_s + 1$ .

#### 2.2.4 Adaptive Pixel-Embedding Strategy dalam PVO-based PEE Embedding

Pengembangan lain dari *PVO-based* dilakukan oleh Weng, et al (2016) yang menggunakan sifat sebuah blok piksel berdasarkan hubungan blok tersebut dengan piksel tetangganya. Langkah pertama pada pengembangan ini adalah membagi citra menjadi blok-blok tidak tumpang tindih (*non-overlapping*) dan berukuran sama satu dengan lainnya. Contoh blok yang dimaksud dapat dilihat pada Gambar 2.5 dengan blok  $P_k$  yang memiliki tetangga piksel  $I_{1,c+1}, I_{2,c+1}, \dots, I_{r+1,c+1}$  dan  $I_{r+1,1}, I_{r+2,2}, \dots, I_{r+1,c}$ .

$I_{1,1}$	$I_{1,2}$	...	$I_{1,c}$	$I_{1,c+1}$
$I_{2,1}$	$I_{2,2}$	...	$I_{2,c}$	$I_{2,c+1}$
...	...	...	...	...
$I_{r,1}$	$I_{r,2}$	...	$I_{r,c}$	$I_{r,c+1}$
$I_{r+1,1}$	$I_{r+2,2}$	...	$I_{r+1,c}$	$I_{r+1,c+1}$

Gambar 2.5 Blok Piksel Berukuran  $r$  Baris dan  $c$  Kolom

Masing-masing blok tersebut kemudian dibedakan berdasarkan persebaran piksel terhadap piksel tetangganya dan dibedakan menjadi 2 jenis yaitu *smooth* dan *complex*. Untuk menentukan jenis blok, Weng, et al (2016) menggunakan persamaan (2.24) sebagai nilai persebaran dan dibandingkan dengan nilai *threshold* ( $vT$ ). Jika lebih dari nilai  $vT$  akan dikelompokkan menjadi *rough*, sebaliknya akan dikelompokkan menjadi *smooth*.

$$\Delta = \sqrt{\frac{\sum_{k \in \{1, \dots, r+1\}} (I_{k,c+1} - u_{LSE})^2 + \sum_{k \in \{1, \dots, c\}} (I_{r+1,k} - u_{LSE})^2 + \sum_{k \in \{4, \dots, n-3\}} (p_{\sigma(k)} - u_{LSE})^2}{r+t+n-5}} \quad (2.24)$$

dengan  $u_{LSE}$  adalah rata-rata dari  $I_{LSE}$  yang beranggotakan piksel tetangga seperti pada Gambar 2.5 dan piksel  $(n - 6)$  dari blok tersebut.

Blok dengan jenis *complex* tidak akan digunakan untuk penyisipan data demi menjaga tingkat kerusakan (*distortion level*) tetap rendah sedangkan blok dengan jenis *smooth* akan dibedakan lagi menjadi 3 tipe untuk penyisipan data. Pembagian tersebut berdasarkan kekuatan korelasi antara piksel dalam blok dengan  $vT$  yang telah ditentukan sebelumnya. Ketiga tipe tersebut adalah tipe *high correlation* dengan nilai  $\Delta \leq \frac{vT}{4}$ , *moderate correlation* dengan kondisi  $\frac{vT}{4} \leq \Delta \leq \frac{vT}{2}$  dan *low correlation* dengan nilai  $\Delta \geq \frac{vT}{2}$ . Setiap tipe korelasi akan dibedakan dalam menentukan nilai *expanded prediction error* ( $PE$ ), baik untuk  $PE$  minimum maupun  $PE$  maksimum. Namun secara garis besar, persamaan untuk menentukan  $PE$  sama yaitu menggunakan persamaan (2.16)-(2.17).

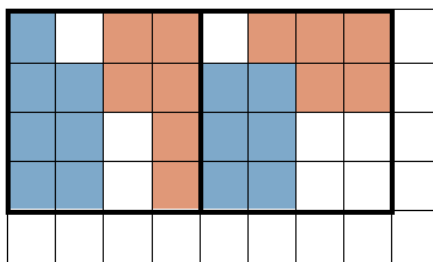
Jika pada persamaan (2.16) menggunakan  $\sigma(n - 1)$  dan  $\sigma(n)$  sebagai indeks piksel penentu nilai  $d_{max}$ , maka pada *high correlation* menggunakan  $\sigma(n), \sigma(n - 1), \sigma(n - 2), \sigma(n - 3)$ . Sedangkan untuk *moderate correlation* menggunakan  $\sigma(n), \sigma(n - 1), \sigma(n - 2)$  dan untuk *low correlation* menggunakan  $\sigma(n), \sigma(n - 1)$ . Begitu pula dalam menentukan nilai  $d_{min}$  dengan persamaan (2.17) menggunakan 4 piksel terkecil, 3 piksel terkecil, dan 2 piksel terkecil untuk masing-masing *high, moderate, dan low correlation*. Selain hal tersebut, persamaan



untuk penyisipan data serta penghitungan nilai piksel baru sama yaitu menggunakan persamaan (2.18)-(2.21).

#### 2.2.5 Pixel Value Grouping (PVG)

*Pixel value grouping* (PVG) merupakan skema pengelompokkan piksel berdasarkan blok dengan ukuran tetap hasil pengembangan dari *pixel value ordering* (He *et al.*, 2016). Gambar 2.6 merupakan contoh dari PVG dengan besar blok piksel adalah 4x4. Awal mula, piksel-piksel dalam blok akan diurutkan (*sorting*) dari piksel berukuran kecil ke besar (*ascending order*). Kemudian akan dipilih nilai terkecil dan terbesar sebagai nilai *reference* dan ditentukan pula nilai *embedding level* (EL).



Gambar 2.6 Pixel Value Grouping (PVG)

Masing-masing piksel dalam blok akan dicari nilai selisihnya untuk ditentukan piksel tersebut dapat bergabung dalam kelompok atau tidak. Batasan piksel tersebut dapat bergabung dalam kelompok adalah nilai selisih dari piksel yang bersangkutan tidak boleh melebihi EL yang telah ditentukan. Hasil dari metode ini adalah kelompok piksel bernilai kecil atau *left pixel group* dan kelompok piksel bernilai besar atau *right pixel group*. Seperti terlihat pada Gambar 2.3, *left pixel group* ditunjukkan dengan warna biru dan *right pixel group* ditunjukkan warna merah. Sedangkan piksel berwarna putih adalah piksel yang tidak memenuhi syarat pengelompokkan sehingga tidak dapat bergabung dalam kelompok manapun dan tidak dapat digunakan dalam proses penyisipan data.

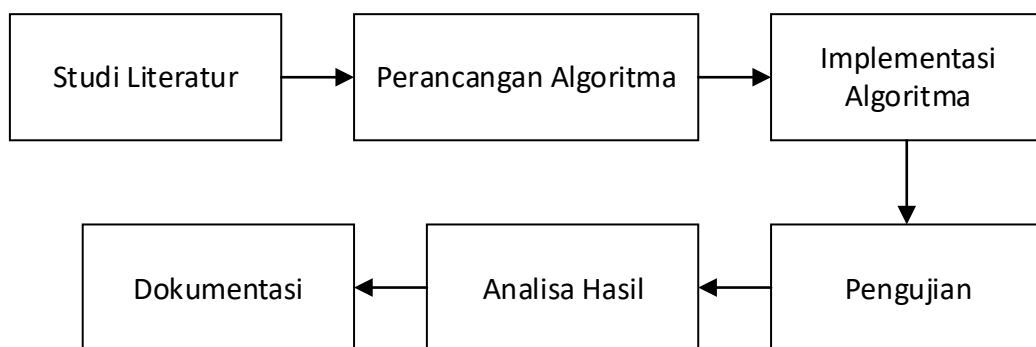
*[Halaman ini sengaja dikosongkan]*

## BAB 3

### METODE PENELITIAN

#### 3.1 Tahapan Penelitian

Tahapan penelitian dibutuhkan agar tujuan yang diharapkan dari penelitian ini dapat tercapai. Langkah awal dari penelitian ini adalah studi literatur seperti yang ditunjukkan pada Gambar 3.1. Studi literatur digunakan untuk mempelajari masalah yang ditemukan dan penelitian saat ini tentang masalah tersebut. Langkah selanjutnya adalah perancangan dan penerapan (implementasi) algoritma. Dalam penelitian ini, algoritma yang dirancang adalah penggunaan *piksel value grouping* yang bersifat adaptif dan proses reduksi *error expansion (REE)* jika data yang disisipkan sebanyak 2 bit dalam satu piksel. Perihal tentang penerapan, algoritma yang telah dirancang akan diterapkan pada citra skala keabuan (*grayscale*). Tahap pengujian dilakukan dengan menyisipkan informasi yang memiliki besar bermacam-macam. Hasil dari tahap pengujian kemudian akan dianalisis untuk mendapatkan kesimpulan dari algoritma yang telah dirancang yang akan ditulis dalam dokumentasi berupa buku tesis.



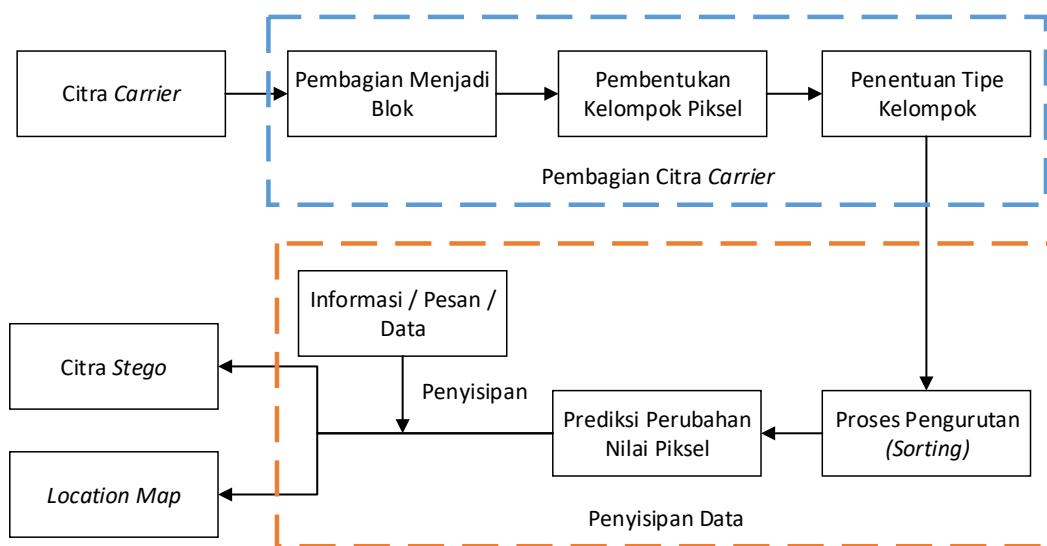
Gambar 3.1 Diagram Alur Penelitian

#### 3.2 Hasil Kajian Tentang *Prediction-Error Expansion*

Penelitian diawali dengan melakukan kajian yang berkaitan dengan topik penelitian yang diajukan. Referensi yang digunakan dalam penelitian ini berasal dari jurnal, konferensi, dan buku yang berkaitan dengan algoritma penyembunyian

data (*data hiding*) dengan media citra digital. Berdasarkan studi literatur tentang *prediction-error expansion* yang telah dilakukan, didapatkan informasi sebagai berikut:

1. Setiap bagian citra memiliki karakteristik berbeda berdasarkan tingkat kehalusan (*smoothness level*) setiap kelompok piksel dalam citra.
2. Penyisipan data pada bagian kelompok piksel dengan tingkat kehalusan yang rendah akan menghasilkan *stego image* dengan tingkat kerusakan lebih besar dibandingkan dengan penyisipan data pada kelompok piksel yang memiliki tingkat kehalusan tinggi.
3. Penggunaan blok pada citra *carrier* belum maksimal, sehingga jumlah data yang disisipkan menjadi terbatas.
4. Belum ditemukan cara untuk menyisipkan data 2 bit sekaligus kedalam sebuah piksel pada konsep penyisipan data lapisan tunggal (*single layer data embedding*).



Gambar 3.2 Alur Perancangan dan Penerapan Algoritma Penyisipan Data

### 3.3 Perancangan Algoritma Penyisipan

Alur perancangan dan penerapan data secara keseluruhan dapat dilihat pada Gambar 3.2. Alur tersebut dibedakan menjadi 4 kelompok tahapan, yaitu : (1) Pembagian Citra *Carrier*, (2) Penyisipan Data, (3) Pembentukan Citra Stego, dan (4) Pembentukan *Location Map*. Secara lebih rinci tahapan-tahapan tersebut dijelaskan pada sub-sub bab dibawah ini.

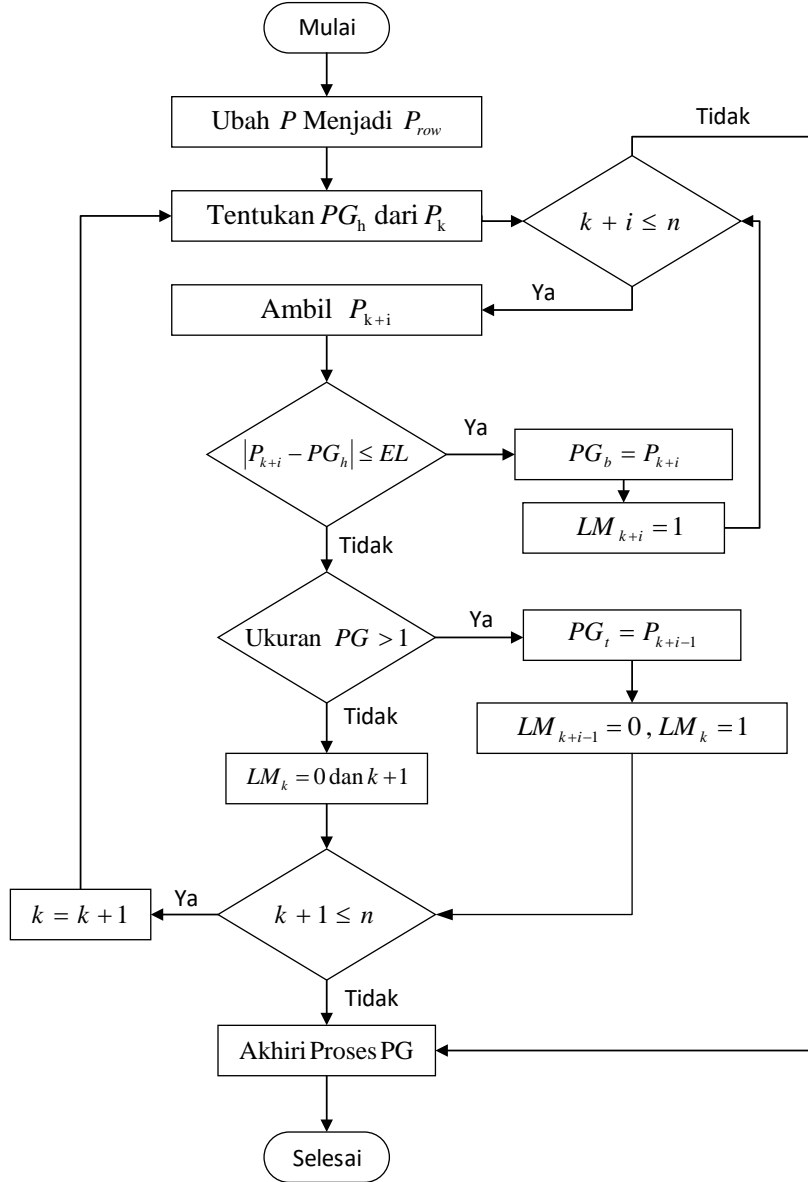
#### 3.3.1 Pembagian Citra *Carrier*

Citra *carrier* yang digunakan untuk penyisipan data akan dibagi menjadi blok-blok dengan ukuran sama dan tidak berpotongan (*non-overlapping block*). Penentuan ukuran dari blok dapat dilihat pada subbab pengujian karena ukuran ini menjadi salah satu atribut dalam pengujian dari algoritma yang dirancang. Setelah dibentuk blok-blok, masing-masing piksel dalam blok akan dikelompokkan dan ditentukan jenis kelompok tersebut sebelum dilakukan prediksi nilai piksel baru.

##### a. Pembentukan Kelompok Piksel

Proses pembentukan kelompok piksel dalam penelitian ini akan dilakukan secara adaptif atau disebut sebagai *adaptive pixel value grouping (APVG)*. Piksel-piksel dalam blok akan dikelompokkan berdasarkan nilai selisihnya dengan *pixel group head* pada setiap kelompok. Proses ini dimulai dengan mengubah blok ( $B$ ) yang berbentuk matrik menjadi matrik baris ( $P_{row}$ ). Kemudian ditentukan *pixel group head* ( $PG_h$ ) sebagai awal mula proses pengelompokkan atau dalam hal ini disebut proses pembentukan *pixel group* ( $PG$ ).  $PG_h$  diambil dari piksel dalam  $P_{row}$  yang ditandai sebagai  $P_k$  dengan  $k$  bernilai 1. Hal ini berarti bahwa  $PG_h$  pertama kali adalah piksel pertama dari masing-masing  $P_{row}$  dari setiap  $B$  yang terbentuk pada citra *carrier* ( $I$ ). Langkah selanjutnya adalah menentukan piksel ( $P_{k+i}$ ) yang dapat digabungkan menjadi satu dengan  $PG_h$  dengan  $i$  bernilai  $\{1, 2, \dots, n\}$  dan  $n$  adalah jumlah maksimum piksel dalam satu blok. Sebuah  $P_{k+i}$  dapat diputuskan

menjadi anggota  $PG$  dalam proses PG bilamana memenuhi persamaan (3.1) dan (3.2).



Gambar 3.3 Diagram Alur Pembentukan Kelompok Pixel

$$d = |P_{k+i} - PG_h| \quad (3.1)$$

$$d \leq EL \quad (3.2)$$

dengan  $EL$  adalah *embedding level* sebagai ambang batas (*threshold*) yang ditentukan sebelumnya oleh pengguna.

Jika persamaan (3.2) terpenuhi maka  $P_{k+i}$  akan dijadikan sebagai *pixel group body* ( $PG_b$ ) dan *location map* -nya ( $LM$ ) diberi nilai 1. Namun jika persamaan tersebut tidak terpenuhi maka akan dilakukan pengecekan jumlah piksel dalam  $PG$ . Apabila jumlah piksel  $PG$  lebih dari satu, piksel yang terakhir bergabung ( $P_{k+i-1}$ ) diubah menjadi *pixel group tail* ( $PG_t$ ) dan  $LM$ -nya menjadi bernilai 0. Proses tersebut akan terus berlanjut hingga  $k + i > n$  dengan  $n = r \times c$ ,  $r$  adalah panjang baris matrik,  $c$  adalah panjang kolom matrik dan proses  $PG$  akan berhenti. Ini berarti bahwa setiap piksel dalam  $P_{row}$  telah diperiksa dalam proses  $PG$ . Hal lain yang mengakibatkan proses  $PG$  berhenti adalah jika setelah penentuan  $PG_h$  tidak terdapat piksel yang menjadi  $PG_b$  atau  $PG_t$  dan nilai dari  $k + 1 > n$ . Kondisi tersebut mencerminkan bahwa  $P_k$  dalam  $P_{row}$  yang akan digunakan sebagai  $PG_h$  telah habis atau tidak terdapat piksel lain dalam sebuah  $P_{row}$ .

Hasil dari proses  $PG$  adalah kelompok-kelompok piksel yang terdiri dari  $PG_h$ ,  $PG_b$ , dan  $PG_t$ . Kelompok ini yang akan diprediksi nilai barunya dan digunakan untuk proses penyisipan data berdasarkan tipe dari kelompok tersebut. Sedangkan piksel yang tidak tergabung dalam  $PG$  akan ditandai sebagai *non-pixel grouping* ( $NPG$ ) dan tidak digunakan untuk penyisipan data. Keseluruhan proses pembentukan kelompok piksel dapat dilihat pada Gambar 3.3 dan contohnya dapat dilihat pada Lampiran 1.

#### b. Penentuan Tipe Kelompok Piksel

Tipe kelompok piksel ditentukan berdasarkan jumlah piksel dalam  $PG$  dibandingkan dengan jumlah piksel dalam blok ( $n$ ). Tipe yang dimaksud dibedakan menjadi tiga, yaitu *smooth*, *rough* dan *other*. Masing-masing tipe kelompok tersebut didapatkan menggunakan persamaan (3.3).

$$tipe\ kelompok\ piksel = \begin{cases} smooth, & \text{jika } \#PG \geq \left(\frac{n}{2}\right) \\ rough, & \text{jika } \left(\frac{n}{2}\right) > \#PG > 4 \\ other, & \text{jika } 4 \geq \#PG \geq 2 \end{cases} \quad (3.3)$$

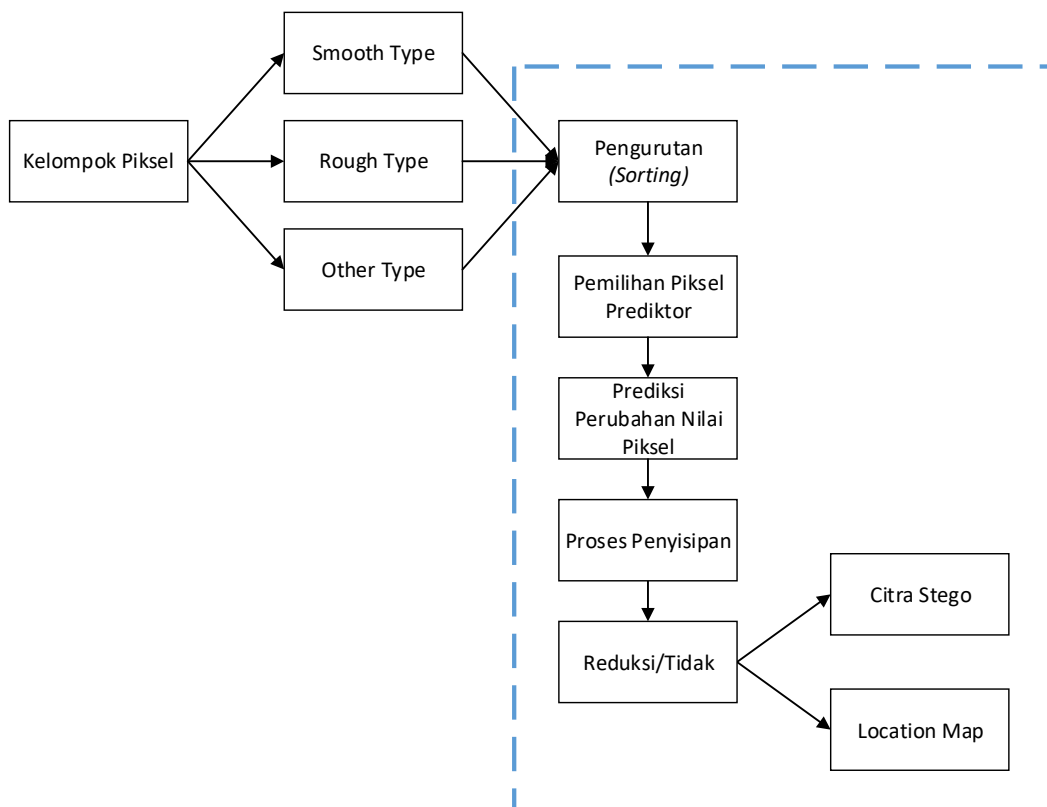
Dengan :

$\#PG$  = jumlah piksel dalam satu kelompok

$n$  = jumlah piksel dalam blok

### 3.3.2 Penyisipan Data

Model prediksi nilai piksel baru dan penyisipan informasi dibedakan menjadi 2 berdasarkan tipe kelompok yang telah ditentukan sebelumnya. Hal yang perlu diperhatikan pada proses penyisipan data adalah data yang disisipkan berjumlah 2 bit untuk 1 piksel. Oleh karena itu, nilai bit data yang kemungkinan disisipkan adalah 00, 01, 10 dan 11. Keseluruhan proses prediksi dan penyisipan data dapat dilihat pada Gambar 3.4.



Gambar 3.4 Proses Penyisipan Data



a. Proses Pengurutan Piksel (*Sorting*) dan Pemilihan Piksel Prediktor

Terlihat pada Gambar 3.4, proses penyisipan data memerlukan kelompok-kelompok piksel. Masing-masing kelompok piksel kemudian dilakukan pengurutan (*sorting*) secara *ascending*. Proses *sorting* ini menghasilkan piksel  $(P_{\sigma(1)}, \dots, P_{\sigma(w)})$ , dengan  $w = \#PG$ ,  $\sigma: \{1, \dots, w\} \rightarrow \{1, \dots, w\}$  adalah pemetaan satu ke satu. Setelah dilakukan proses *sorting*, proses dilanjutkan dengan melakukan pemilihan 2 nilai referensi (*ref*) yaitu nilai referensi minimum ( $ref_{\min}$ ) dan nilai referensi maksimum ( $ref_{\max}$ ). Nilai  $ref_{\min}$  digunakan sebagai piksel prediktor minimum sedangkan nilai  $ref_{\max}$  digunakan sebagai piksel prediktor maksimum. Seperti yang telah dijelaskan sebelumnya, nilai *ref* dibedakan berdasarkan  $\#PG$  dan tipe blok. Berikut cara pemilihan nilai *ref* baik nilai  $ref_{\min}$  maupun nilai  $ref_{\max}$ :

- Tipe *Smooth*

$$ref_{\max} = \begin{cases} \frac{\#PG}{2} + 1, & \text{jika } \#PG \bmod 2 = 0 \\ \#PG/2, & \text{jika } \#PG \bmod 2 = 1 \end{cases} \quad (3.4)$$

$$ref_{\min} = \#PG/2 \quad (3.5)$$

- Tipe *Rough*

$$ref_{\max} = \begin{cases} \frac{\#PG}{2} + 2, & \text{jika } \#PG \bmod 2 = 0 \\ \frac{\#PG}{2} + 1, & \text{jika } \#PG \bmod 2 = 1 \end{cases} \quad (3.6)$$

$$ref_{\min} = \frac{\#PG}{2} - 1 \quad (3.7)$$

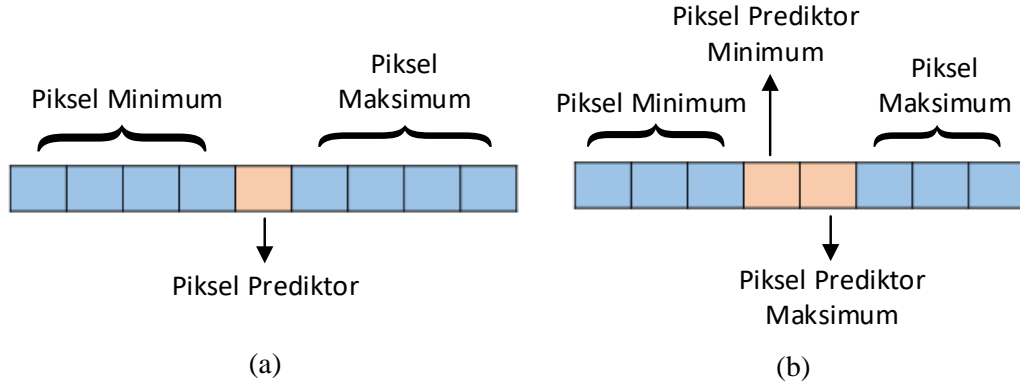
- Tipe *Other*

$$ref_{\max} = \begin{cases} 3, & \text{jika } \#PG = 4 \\ 2, & \text{jika } \#PG = 3 \\ 1, & \text{jika } \#PG = 2 \end{cases} \quad (3.8)$$

$$ref_{\min} = 2, \text{ jika } \#PG \in \{4, 3\} \quad (3.9)$$

dengan  $\#PG \bmod 2$  adalah fungsi modular 2 dari  $\#PG$ . Persamaan (3.4)-(3.9) merupakan hasil pengembangan dari metode yang diusulkan Weng, et al (2016).

Pengembangan ini bertujuan agar ketersediaan piksel untuk penyisipan data lebih besar dibandingkan skema milik Weng, et al (2016). Hal ini memberikan pengaruh untuk kapasitas informasi yang dapat disisipkan akan bertambah pula. Contoh penentuan piksel prediktor untuk tipe *smooth* dapat dilihat pada Gambar 3.5.



Gambar 3.5 Piksel Prediktor pada Tipe *Smooth*, (a) Piksel Group dengan Anggota Piksel Berjumlah Ganjil, (b) Piksel Group dengan Anggota Piksel Berjumlah Genap

#### b. Prediksi Perubahan Nilai Piksel

Setelah didapatkan nilai *ref* langkah selanjutnya adalah menghitung nilai perubahan piksel atau *expanded prediction error* ( $d$ ). Jika pada persamaan (2.16)-(2.17) nilai  $\sigma(n - 1)$  sebagai indeks piksel prediktor, maka pada penelitian ini nilai tersebut diubah menjadi  $\sigma(ref_{max})$ . Sedangkan untuk  $\sigma(2)$  diubah menjadi  $\sigma(ref_{min})$ . Hasil dari penyesuaian persamaan tersebut adalah sebagai berikut:

$$d_{max} = P_u - P_v, \begin{cases} u = \min(\sigma(n), \sigma(ref_{max})) \\ v = \max(\sigma(n), \sigma(ref_{max})) \end{cases} \quad (3.10)$$

$$d_{min} = P_s - P_t, \begin{cases} s = \min(\sigma(i), \sigma(ref_{min})) \\ t = \max(\sigma(i), \sigma(ref_{min})) \end{cases} \quad (3.11)$$

Dengan :

$P_u, P_v, P_s, P_t$  = piksel dalam  $P_{row}$

$\sigma(n)$  = indeks piksel yang bernilai lebih besar dari piksel prediktor maksimum

$\sigma(ref_{max})$  = indeks piksel prediktor maksimum

$\sigma(i)$  = indeks piksel yang bernilai lebih kecil dari piksel prediktor minimum

$\sigma(ref_{min})$  = indeks piksel prediktor minimum

### c. Penyisipan Data atau Informasi

Persamaan yang digunakan untuk menyisipkan data, sama dengan persamaan yang digunakan oleh Peng et al (2014) yaitu persamaan (2.18)-(2.19) dengan  $b$  adalah nilai pesan yang diubah menjadi nilai desimal. Kemungkinan nilai  $b$  yang terbentuk pada penelitian ini adalah 00, 01, 10 dan 11 dan jika diubah kedalam nilai desimal masing-masing adalah 0, 1, 2, dan 3. Proses penyisipan tersebut akan membuat nilai perubahan piksel menjadi lebih besar dari sebelumnya. Oleh karena itu, penelitian ini juga mengusulkan metode untuk mengurangi nilai tersebut yang disebut reduksi *error expansion* atau *reduction error expansion* (*REE*). Proses reduksi nilai ini dapat diselesaikan dengan menggunakan persamaan (3.12)-(3.13).

$$d''_{max} = \begin{cases} d'_{max} \bmod 2 & , \text{jika } |d'_{max}| > 1 \text{ dan } b = \{10, 11\} \\ d'_{max} & , \text{sebaliknya} \end{cases} \quad (3.12)$$

$$d''_{min} = \begin{cases} d'_{min} \bmod 2 & , \text{jika } |d'_{min}| > 1 \text{ and } b = \{10, 11\} \\ d'_{min} & , \text{sebaliknya} \end{cases} \quad (3.13)$$

dengan  $d''_{max}$  dan  $d''_{min}$  masing-masing merupakan nilai *expanded prediction error* yang baru. Proses reduksi ini bertujuan untuk menjaga agar tingkat kerusakan dari *carrier* tidak terlalu besar.

### 3.3.3 Pembuatan Citra *Stego*

Tahap akhir dari algoritma yang dirancang adalah pembuatan citra *stego*. Citra ini dapat dibuat dengan mengganti piksel lama ( $P$ ) dengan piksel baru ( $P'$ ) yang telah disisipi data atau informasi. Berbeda dengan pengembangan PEE sebelumnya, penghitungan  $P'$  dalam metode ini dibedakan menjadi 2, yaitu  $P'$  dengan reduksi *error expansion* dan  $P'$  tanpa reduksi *error expansion*. Nilai  $P'$  hasil dari penyisipan data tanpa reduksi didapatkan menggunakan persamaan (3.14)-(3.15). Sedangkan nilai  $P'$  dari hasil reduksi didapatkan dengan menggunakan persamaan (3.16)-(3.17).

$$P'_{\sigma(i)} = P_{\sigma(ref_{min})} - |d''_{min}| \quad (3.14)$$

$$P'_{\sigma(j)} = P_{\sigma(ref_{max})} + |d''_{max}| \quad (3.15)$$

$$P'_{\sigma(i)} = P_{\sigma(i)} - |d''_{min}| \quad (3.16)$$

$$P'_{\sigma(j)} = P_{\sigma(j)} + |d''_{max}| \quad (3.17)$$

Dengan :

$P'_{\sigma(i)}$  = piksel baru ( $P'$ ) dari piksel minimum

$P'_{\sigma(j)}$  = piksel baru ( $P'$ ) dari piksel maksimum

$P_{\sigma(i)}$  = piksel lama ( $P$ ) dari piksel minimum

$P_{\sigma(j)}$  = piksel lama ( $P$ ) dari piksel maksimum

$P_{\sigma(ref_{min})}$  = piksel referensi untuk piksel minimum

$P_{\sigma(ref_{max})}$  = piksel referensi untuk piksel maksimum

### 3.3.4 *Location Map (LM)*

*Location map (LM)* dalam konsep *data hiding* digunakan untuk menyimpan informasi mengenai keadaan suatu piksel seperti piksel tersebut mengakibatkan *overflow* atau *underflow*. Berdasarkan kebutuhan usulan metode ini, *location map* dibagi menjadi tiga bit dengan fungsi yang berbeda. Bit pertama

digunakan untuk mengenali kelompok piksel, bit kedua menandai piksel yang mengalami *overflow* dan *underflow* dan bit ketiga untuk menunjukkan bahwa piksel tersebut mengalami reduksi. Masing-masing bagian tersebut memiliki dua nilai bit yang berbeda yaitu 1 dan 0. Penggunaan *location map* dalam usulan metode ini dapat dilihat pada Tabel 3.1.

Tabel 3.1 Penggunaan Location Map

	Kelompok Piksel		<i>Overflow/Underflow</i>		Reduksi <i>Error Expansion</i>	
	$PG_h, PG_b$	$PG_t, NPG$	Ya	Tidak	Ya	Tidak
Nilai <i>LM</i>	1	0	1	0	1	0

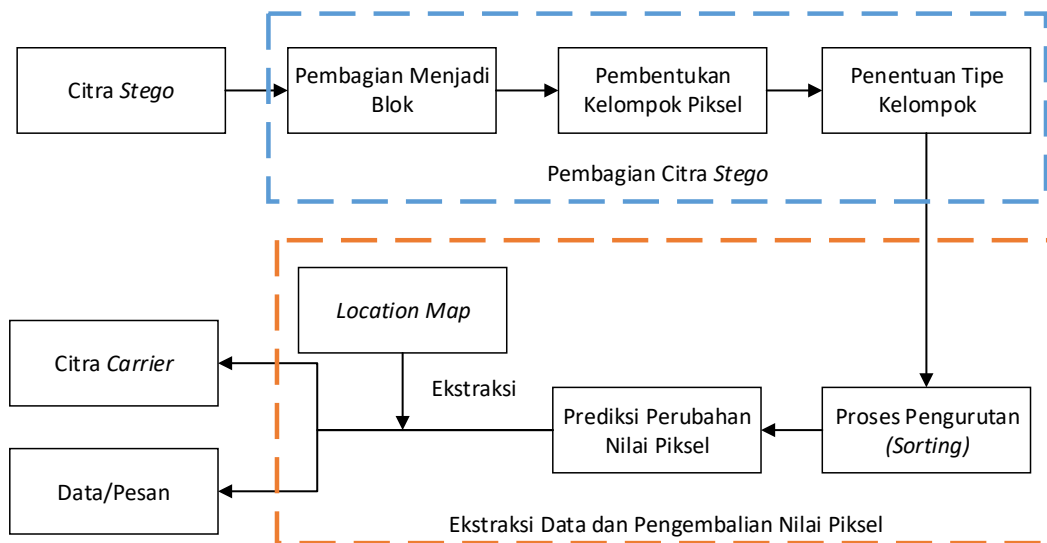
Nilai *location map* yang digunakan untuk menunjuk kelompok piksel memiliki nilai 1 jika piksel tersebut merupakan *pixel group head* ( $PG_h$ ) atau *pixel group body* ( $PG_b$ ). Sebaliknya akan bernilai 0 jika piksel tersebut merupakan *pixel group tail* ( $PG_t$ ) atau *non pixel grouping* (NPG). Sedangkan untuk bit kedua dan ketiga, nilai 1 menunjukkan bahwa piksel tersebut mengalami *overflow/underflow* setelah proses penyisipan data dan mengalami reduksi *error expansion*. Jika bernilai 0, piksel tersebut tidak mengalami *overflow/underflow* dan tidak mengalami reduksi *error expansion*. Bagian-bagian tersebut kemudian disatukan sehingga hanya terdapat satu *file location map* untuk setiap kali proses penyisipan pada sebuah citra.

Agar data dalam citra *stego* bisa didapatkan kembali, *location map* ini perlu disertakan dalam proses ekstraksi data dan pengembalian citra *carrier*. Dalam metode Li, et al. (2013), Peng, et al. (2014) dan Weng, et al. (2015), *location map* disisipkan dalam citra *carrier* menggunakan konsep *lossless compression*. Namun pada usulan metode ini, *location map* tidak akan disisipkan dalam citra *carrier*. Hal ini dilakukan karena jika *location map* disisipkan dalam citra *carrier* akan

memperbesar tingkat kerusakan citra *carrier* yang sebelumnya telah disisipi data (Al Huti et al., 2015).

### 3.4 Perancangan Algoritma Ekstraksi

Sama seperti proses penyisipan data, proses ekstra data dimulai dengan membagi citra *stego* menjadi blok dengan ukuran yang sama. Selanjutnya dibentuk kelompok piksel atau *pixel group (PG)* berdasarkan *location map (LM)* yang diterima. Jika *PG* telah dibentuk, masing-masing *PG* tersebut akan dilakukan penentuan tipe kelompok dan pengurutan atau *sorting* secara *ascending*. Langkah selanjutnya adalah prediksi perubahan nilai piksel dan ekstraksi data atau pesan. Tahap terakhir adalah pembentukan citra *carrier* hasil dari pengembalian nilai piksel asli. Keseluruhan proses ini ditunjukkan oleh Gambar 3.6.

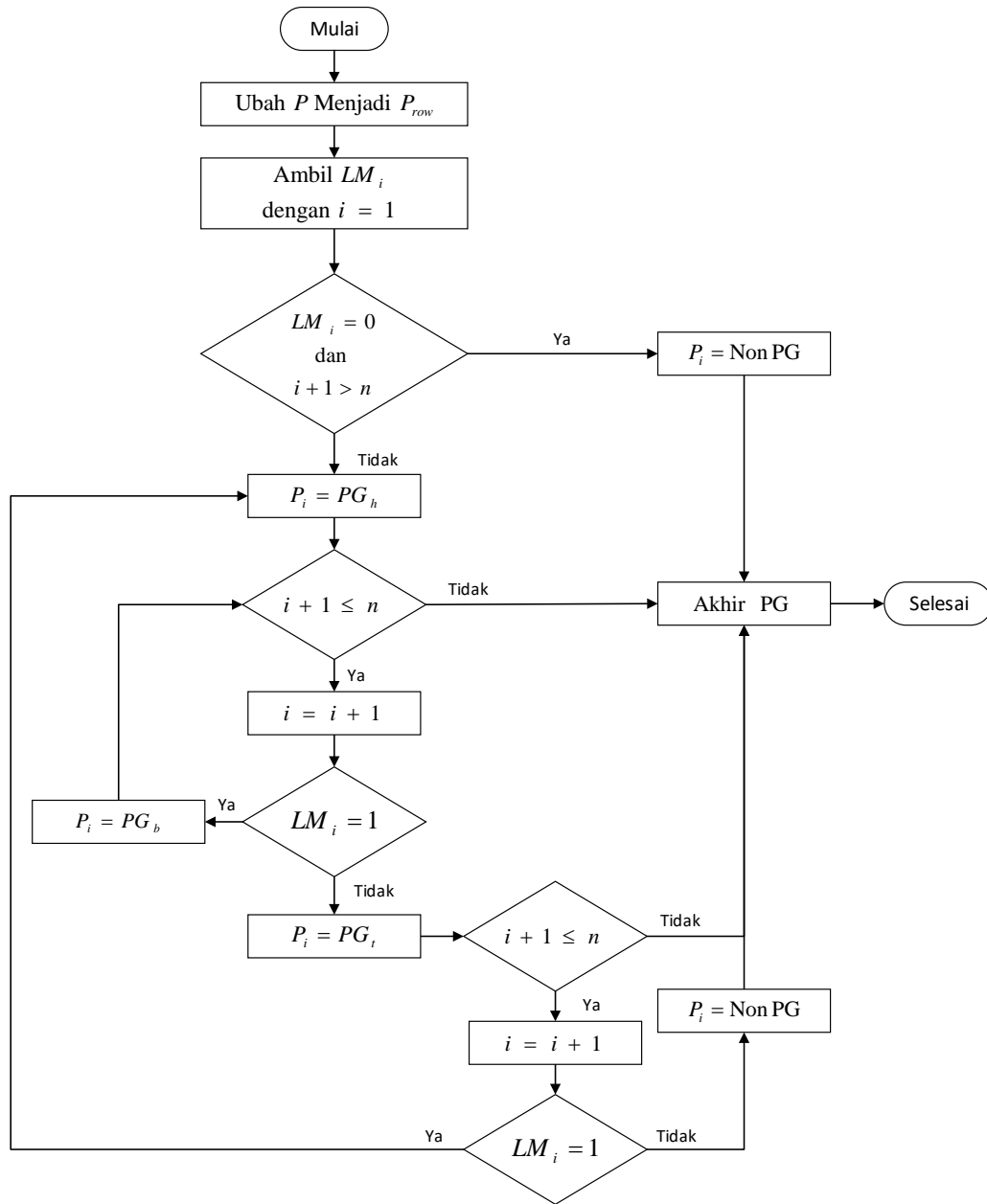


Gambar 3.6 Alur Ekstraksi Data dan Pengembalian Nilai Piksel

#### 3.4.1 Pembagian Citra Stego

Tahap ini membagi *citra stego* menjadi blok-blok dengan ukuran yang sama saat proses penyisipan. Blok yang dibentuk merupakan blok yang tidak beririsan (*non-overlapping*). Masing-masing blok (*P*) akan diproses lebih lanjut

untuk membentuk kelompok piksel atau *pixel group* (*PG*) berdasarkan *location map* (*LM*) yang telah disediakan. Pembentukan *PG* secara keseluruhan dapat dilihat pada Gambar 3.7.



Gambar 3.7 Diagram Alur Pembentukan Kelompok Piksel (*PG*)

Langkah pertama dalam pembentukan adalah mengubah  $P$  yang semula berbentuk matrik menjadi matrik baris ( $P_{row}$ ). Kemudian, pembentukan kelompok piksel ( $PG$ ) dimulai berdasarkan  $LM$  yang didapat. Piksel dalam  $P_{row}$  ( $P_i$ ) akan menjadi *pixel group head* ( $PG_h$ ) jika memenuhi persamaan (3.18). Sedangkan untuk *pixel group body* ( $PG_b$ ) dan *pixel group tail* ( $PG_t$ ), masing-masing terbentuk jika memenuhi persamaan (3.19) dan (3.20). Diluar persamaan-persamaan yang disebutkan, maka  $P_i$  tidak dikelompokkan kedalam  $PG$  dan menjadi *non-pixel group* ( $NPG$ ).

$$PG_h = P_i, \text{ jika } LM_i = 1 \text{ dan } i + 1 \leq n \quad (3.18)$$

$$PG_b = P_i, \text{ jika } LM_i = 1, LM_{i-1} = 1 \quad (3.19)$$

$$PG_t = P_i, \text{ jika } LM_i = 0, LM_{i-1} = 1 \quad (3.20)$$

dengan  $i$  adalah  $1, 2, \dots, n$  dan  $n$  merupakan jumlah piksel dalam satu blok.

#### 3.4.2 Pengembalian Nilai Piksel dan Ekstraksi Data

Pengembalian nilai piksel dan ekstraksi data membutuhkan piksel prediktor yang didapatkan setelah kelompok piksel terbentuk. Sama halnya dengan proses penyisipan data, kelompok piksel pada ekstraksi pesan juga dibedakan menjadi 3, yaitu untuk tipe *smooth*, *rough* serta *other* yang masing-masing piksel prediktornya bisa didapatkan menggunakan persamaan (3.4)-(3.9). Setelah piksel prediktor didapatkan, maka dapat dilakukan penghitungan nilai  $d''_{max}$  dan  $d''_{min}$  menggunakan persamaan (3.21)-(3.22).

$$d''_{max} = P'_u - P'_v \begin{cases} u = \min(\sigma(n), \sigma(ref_{max})) \\ v = \max(\sigma(n), \sigma(ref_{max})) \end{cases} \quad (3.21)$$

$$d''_{min} = P'_s - P'_t \begin{cases} s = \min(\sigma(i), \sigma(ref_{min})) \\ t = \max(\sigma(i), \sigma(ref_{min})) \end{cases} \quad (3.22)$$



Dengan :

$d''_{max}$  = selisih piksel prediktor dengan piksel maksimum untuk penyisipan data

$d''_{min}$  = selisih piksel prediktor dengan piksel minimum untuk penyisipan data

$P'_u, P'_v, P'_s, P'_t$  = piksel dalam  $P_{row}$

$\sigma(n)$  = indeks piksel yang bernilai lebih besar dari piksel prediktor maksimum

$\sigma(ref_{max})$  = indeks piksel prediktor maksimum

$\sigma(i)$  = indeks piksel yang bernilai lebih kecil dari piksel prediktor minimum

$\sigma(ref_{min})$  = indeks piksel prediktor minimum

Data yang disisipkan bisa didapatkan kembali berdasarkan *location map* yang menandakan bahwa piksel tersebut mengalami reduksi *error expansion* atau tidak. Hal ini juga berlaku dalam pengembalian nilai piksel asli untuk pembangunan citra *carrier*. Persamaan dibawah ini dapat digunakan untuk mendapatkan data yang disisipkan dan piksel tersebut mengalami reduksi *error expansion*.

$$b = \begin{cases} 3, & \text{jika } d''_{max} = \{-1,1\} \\ 2, & \text{jika } d''_{max} = \{0,2\} \end{cases} \quad (3.23)$$

$$b = \begin{cases} 3, & \text{jika } d''_{min} = \{-1,1\} \\ 2, & \text{jika } d''_{min} = \{0,2\} \end{cases} \quad (3.24)$$

dengan  $b$  adalah pesan yang disisipkan. Sedangkan untuk proses pengembalian piksel menggunakan persamaan berikut :

$$d_{max} = \begin{cases} d''_{max} - 1, & \text{jika } d''_{max} > 0 \\ -d''_{max}, & \text{jika } d''_{max} \leq 0 \end{cases} \quad (3.25)$$

$$P_{(j)} = P'_{(j)} - d_{max} \quad (3.26)$$

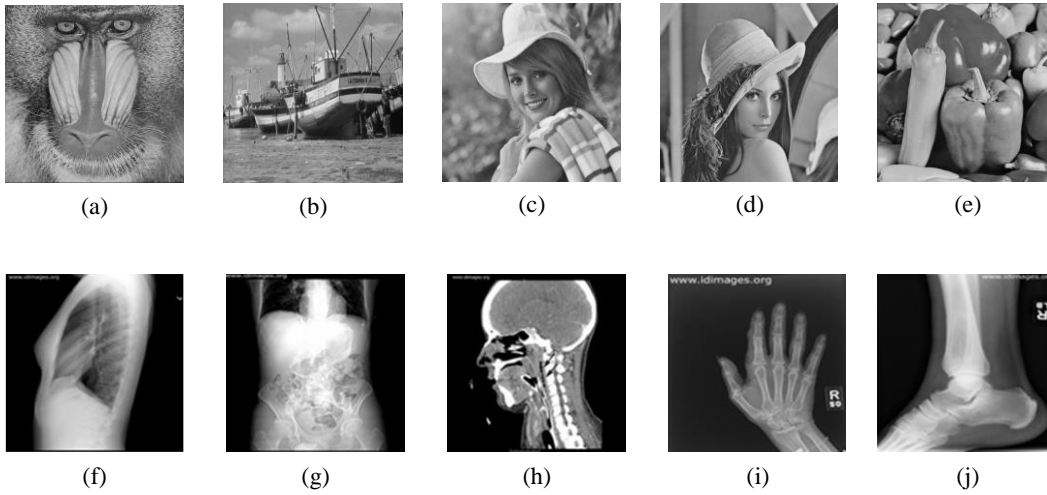
$$d_{min} = \begin{cases} d''_{min} - 1, & \text{jika } d''_{min} > 0 \\ -d''_{min}, & \text{jika } d''_{min} \leq 0 \end{cases} \quad (3.27)$$

$$P_{(i)} = P'_{(i)} + d_{min} \quad (3.28)$$

dengan  $P'_{(i)}$  dan  $P'_{(j)}$  adalah piksel dari citra *stego* sedangkan  $P_{(i)}$  dan  $P_{(j)}$  adalah piksel yang bernilai asli. Piksel citra *stego* yang tidak mengalami reduksi, dapat menggunakan konsep dari Peng, et al. (2014) untuk mendapatkan data yang disisipkan dan mengembalikan nilai piksel asli.

### 3.4.3 Pembangunan Citra *Carrier*

Pembangunan citra *carrier* adalah tahap terakhir dari proses ekstraksi pesan. Citra ini dihasilkan dengan mengganti piksel yang telah disisipi pesan ( $P'_{(i)}$  atau  $P'_{(j)}$ ) dengan piksel aslinya ( $P_{(i)}$  atau  $P_{(j)}$ ). Proses penghitungan nilai piksel asli dibedakan menjadi dua cara, yaitu penghitungan dari hasil reduksi *error expansion* dan penghitungan dari proses penyisipan biasa. Masing-masing cara tersebut telah dijelaskan pada sub bab sebelumnya.



Gambar 3.8 Citra *Carrier* : (a) Baboon, (b) Boat, (c) Elaine, (d) Lena, (e) Peppers, (f) Abdomal, (g) Chest, (h) Head, (i) Hand, (j) Leg

### 3.5 Rencana Implementasi Algoritma

Implementasi dari metode ini menggunakan perangkat lunak Matlab versi R2017a dengan *carrier* citra skala abu-abu yang didapatkan dari *USC-SIPI image database* dan *Partners Healthcare System database*. Daftar citra yang digunakan dapat dilihat pada Gambar 3.8. Masing-masing citra memiliki ukuran yang sama yaitu 512x512 piksel. Citra ini dibedakan menjadi 2 kelompok, yaitu kelompok citra *grayscale* dan kelompok citra medis. Kelompok citra *grayscale* ditunjukkan Gambar 3.8(a)-3.8(e), sedangkan kelompok citra medis ditunjukkan Gambar 3.8(f)-3.8(j). Penggunaan citra pada kelompok *grayscale* sama dengan citra yang digunakan pada penelitian Peng et al (2014), Weng et al (2016) dan He et al (2016).

### 3.6 Pengujian

Pengujian dilakukan dengan mengukur tingkat kerusakan yang didapatkan *carrier* setelah pesan disisipkan. Pengukuran tingkat kerusakan menggunakan fungsi matematis dari *peak signal to noise ratio* (PSNR). Jika nilai PSNR semakin tinggi, ini berarti tingkat kerusakan yang dihasilkan metode penyisipan semakin kecil. Sebaliknya, jika nilai PSNR semakin rendah, maka tingkat kerusakan yang dihasilkan juga semakin besar (Wu, 2002). Pengujian akan menggunakan beberapa skema untuk mengetahui pengaruh dari metode yang diusulkan dengan tingkat kerusakan dan kapasitas dari citra *stego*. Pengujian yang dimaksud dibedakan menjadi dua, yaitu :

#### a. Pengujian kapasitas citra *stego*

Pengujian kapasitas dilakukan dengan menyisipkan data biner sebanyak piksel untuk penyisipan data. Jumlah dari data yang disisipkan kemudian dihitung untuk mengetahui kapasitas dari citra *stego* yang dihasilkan. Hasil ini kemudian dibandingkan untuk mengetahui besar kapasitas dari metode yang diusulkan dengan metode referensi.

#### b. Pengujian kualitas citra *stego*

Pengujian ini dilakukan dengan menyisipkan pesan yang memiliki ukuran berbeda serta *embedding level* yang bervariasi. Ukuran pesan yang digunakan

mengikuti skema uji coba yang dilakukan oleh Peng et al (2014) dan Weng et al (2016). Sedangkan nilai untuk *embedding level* didapatkan dari He et al (2016). Berbeda dengan nilai untuk ukuran blok yang merupakan usulan pengujian sendiri untuk mengetahui pengaruh besar blok dengan tingkat kerusakan citra stego dari metode yang diusulkan. Keseluruhan nilai dari variabel uji coba data dilihat pada Tabel 3.2.

Tabel 3.2 Skema Pengujian

Variabel	Nilai
Ukuran Pesan	1 Kb – 100 Kb
<i>Embedding Level</i> ( <i>EL</i> )/ <i>Threshold</i>	1-20
Ukuran Blok	3x3 – 10x10

### 3.7 Analisa Hasil

Analisa digunakan untuk mengetahui capaian yang dihasilkan dari metode yang dirancang. Capaian ini dapat dilihat berdasarkan perbandingan hasil pengujian yang dilakukan dengan hasil uji dari metode yang terkait. Perbandingan tersebut memiliki syarat yang harus dipenuhi agar hasil evaluasi sah atau valid, yaitu *carrier*, jenis dan panjang pesan serta ukuran yang digunakan harus sama. Hal yang dibandingkan adalah kualitas dan kapasitas dari citra *stego*. Kualitas ini didapatkan dengan menerapkan metode pengukuran tingkat derau (*noise*) yaitu *peak signal to noise ratio (PSNR)* pada citra *stego* yang dimaksud. Sedangkan kapasitas didapatkan dengan menyisipkan data sebanyak mungkin hingga piksel untuk penyisipan telah disisipi data secara keseluruhan.

## BAB 4

### HASIL DAN PEMBAHASAN

#### 4.1 Lingkungan Uji Coba dan Implementasi Algoritma

Subbab ini menjelaskan tentang hasil implementasi algoritma yang berbentuk *pseudo-code* dan lingkungan uji coba yang digunakan dalam penelitian ini. Penjelasan akan dibagi menjadi beberapa bagian, yaitu: lingkungan uji coba yang digunakan, implementasi *adaptive pixel value grouping* (APVG), dan implementasi reduksi *error expansion* (REE).

##### 4.1.1 Lingkungan Uji Coba

Lingkungan uji coba pada implementasi algoritma ini meliputi perangkat keras dan perangkat lunak. Perangkat keras yang dimaksud adalah jenis *processor* dan *memory* (RAM) yang digunakan, sedangkan perangkat lunak adalah sistem operasi serta *tools* untuk pengujian. Rincian tersebut dapat dilihat pada Tabel 4.1 dibawah ini.

Tabel 4.1 Rincian Lingkungan Uji Coba

Jenis Lingkungan	Rincian	
Perangkat Keras	<i>Processor</i>	Intel Core i5-7400 3.00 Ghz
	<i>Memory (RAM)</i>	8192 MB
Perangkat Lunak	Sistem Operasi	Windows 10 Pro 64-bit
	<i>Tools</i>	Matlab 2017a

##### 4.1.2 Implementasi *Adaptive Pixel Value Grouping*

Hasil perancangan algoritma dari *adaptive pixel value grouping* (APVG) pada subbab 3.3.1 kemudian diolah menjadi *pseudo-code*. *Pseudo-code* ini akan diterapkan pada *tools* untuk pengujian, dalam hal ini adalah *Matlab*. Gambar 4.1 menunjukkan *pseudo-code* untuk algoritma APVG.

```

(1) Raster scan from  $B$  to obtain  $\{P_1, P_2, \dots, P_n\}$ 
(2) Initiate the  $EL$ ,  $PG_h$ ,  $PG$  and  $PGList$ 
    (a) Set  $EL = \text{input user}$ 
    (b) Set  $PG_h = P_1$ 
    (c) Set  $PG = 0$ 
    (d) Set  $PGList = 0$ 
(3) Start grouping process
    (a) For  $i = 1:n$ 
        If  $i == 1$ 
             $PG.add(PG_h)$ 
        Else If  $-EL \leq P_i - PG_h \leq EL$ 
             $PG.add(P_i)$ 
        Else If  $PG.element() > 1$ 
             $PGList.add(PG)$ 
            set  $PG_h = P_i$ 
             $PG.add(PG_h)$ 
        Else
            set  $PG_h = P_i$ 
             $PG.add(PG_h)$ 
        end
    (b) end looping

```

Gambar 4.1 *Pseudo-code* Algoritma Adaptive Pixel Value Grouping (APVG)

#### 4.1.3 Implementasi Reduksi Error Expansion

Algoritma Reduksi Error Expansion (REE) digunakan pada saat melakukan proses penyisipan data. Setelah dilakukan perancangan algoritma REE pada subbab 3.3.2 bagian (c), maka dibuat *pseudo-code* agar algoritma ini dapat diterapkan dalam *tools*. *Pseudo-code* tersebut dapat dilihat pada Gambar 4.2 yang menunjukkan *pseudo-code* proses penyisipan data serta penerapan algoritma REE pada bagian minimum dalam *pixel group*. Sedangkan untuk penerapan algoritma REE pada bagian maksimum kurang lebih sama, hanya proses penghitungan nilai *expanded prediction error* dan nilai piksel baru yang membedakan keduanya.

```

(1) Initiate the  $P_i$ ,  $P_{ref}$ , and  $b$ 
    (a) Set  $P_i$  = pixel for embedding in  $PG$ 
    (b) Set  $P_{ref}$  = pixel predictor
    (c) Set  $b$  = data to embed
(2) Calculate error expansion value ( $d$ )
    (a)  $x = findMin(P_i.index(), P_{ref}.index())$ 
    (b)  $y = findMax(P_i.index(), P_{ref}.index())$ 
    (c)  $d = x - y$ 
(3) Start embedding process in minimum part
    (a) If  $d == 1$ 
         $d' = d + decimalForm(b)$ 
        If  $|d'| > 1$ 
            If  $decimalForm(b) \in \{2,3\}$ 
                 $d'' = d'.mod(2)$ 
                 $P'_i = P_i - |d''|$ 
            Else
                 $d'' = d'$ 
                 $P'_i = P_{ref} - |d''|$ 
            End
        Else
             $d'' = d'$ 
             $P'_i = P_{ref} - |d''|$ 
        End
    (b) If  $d == 0$ 
         $d' = d - decimalForm(b)$ 
        If  $|d'| > 1$ 
            If  $decimalForm(b) \in \{2,3\}$ 
                 $d'' = d'.mod(2)$ 
                 $P'_i = P_i - |d''|$ 
            Else
                 $d'' = d'$ 
                 $P'_i = P_{ref} - |d''|$ 
            End
        Else
             $d'' = d'$ 
             $P'_i = P_{ref} - |d''|$ 
        End
    (c) If  $d > 1$ 
         $d' = d + 1$ 
         $P'_i = P_i - |d'|$ 
    (d) If  $d < 0$ 
         $d' = d - 1$ 
         $P'_i = P_i - |d'|$ 

```

Gambar 4.2 Pseudo-code Algoritma *Reduksi Error Expansion* pada Proses Penyisipan di Bagian Minimum

## 4.2 Kualitas dan Kapasitas Citra Stego

Kapasitas dan kualitas citra *stego* yang dihasilkan oleh metode usulan akan dibandingkan dengan metode dari Peng et al (2014). Perbandingan ini dilakukan untuk mengetahui kemampuan metode usulan dengan metode referensi yaitu Peng et al (2014). Hasil dari perbandingan kualitas citra *stego* dapat dilihat pada Tabel 4.2 yang menunjukkan kualitas citra *stego* dari metode Peng, metode usulan APVG dan metode usulan APVG dengan reduksi *error expansion (REE)* pada blok berukuran 4x4 dan besar data 1 Kb.

Beberapa citra *carrier* pada metode usulan (APVG dengan REE) memiliki kualitas lebih baik dibandingkan dengan metode Peng (2014). Misal pada citra 'Lena' meningkat sebesar 4,58 dB dari 71,57 dB menjadi 76,15 dB. Hal ini terjadi karena jumlah *shifted pixel* berkurang secara signifikan dari 699 menjadi 169. Berkurangnya jumlah *shifted pixel* ini memberikan efek terhadap kualitas citra *stego* dari metode usulan. Keadaan ini juga terjadi ketika metode APVG tidak diterapkan bersama metode REE. Walaupun terjadi penurunan kualitas citra *stego*, namun penurunan tersebut tidak terlalu banyak dan diimbangi dengan meningkatnya kapasitas citra *stego* yang dapat dilihat pada Tabel 4.3.

Tabel 4.2 Perbandingan Kualitas Citra Stego

<i>Carrier</i>	PSNR (dB)			<i>Shifted Pixel</i>		
	Peng (2014)	APVG	APVG + REE	Peng (2014)	APVG	APVG + REE
Baboon	67,93	69,33	75,56	2255	246	246
Boat	71,01	69,47	76,04	864	185	185
Elaine	72,36	69,76	77,49	503	59	59
Lena	71,57	69,50	76,15	699	169	169
Pepper	70,29	69,42	75,82	1107	206	206
Abdomal	75,13	71,47	78,39	35	93	93
Chest	75,03	70,77	77,62	47	111	111
Hand	75,37	69,66	76,99	7	102	102
Head	74,76	72,22	79,37	82	69	69
Leg	75,38	69,82	77,97	6	34	34



Terlihat pada Tabel 4.3. bahwa peningkatan kualitas terjadi secara signifikan untuk semua citra *carrier* dengan blok yang digunakan berukuran 4x4. Rata-rata kapasitas dari citra *stego* meningkat sebesar 156.776 bit dengan penurunan kualitas rata-rata sebesar 8,317 dB. Jika dilihat lebih seksama, kelompok citra medis memberikan peningkatan yang paling besar dibandingkan dengan kelompok citra *grayscale*. Hal ini karena pada kelompok citra medis memiliki area *smooth* lebih banyak dibandingkan dengan kelompok citra *grayscale* seperti ‘Baboon’. Kondisi tersebut sesuai dengan penjelasan dari Al-Qershi (2011) dan Yang (2016). Disisi lain, peningkatan kualitas dan kapasitas terjadi karena pada metode usulan mampu memanfaatkan blok dengan tipe *rough*. Tipe ini pada metode Peng tidak digunakan untuk proses penyisipan data karena akan menyebabkan kualitas citra *stego* menurun.

Tabel 4.3 Perbandingan Kapasitas

Citra <i>Carrier</i>	Kapasitas (bit)		PSNR (dB)	
	Metode Usulan	Peng (2014)	Metode Usulan	Peng (2014)
Baboon	31400	5300	48,86	57,52
Boat	65160	9582	49,04	57,85
Elaine	65520	8946	49,03	57,79
Lena	95696	12762	49,19	58,11
Pepper	81974	11035	49,12	57,96
Abdomal	235806	14121	50,52	60,38
Chest	289170	13723	51,09	61,26
Hand	299628	26519	59,43	59,42
Head	259940	8877	51,02	60,92
Leg	281190	26836	50,33	59,59

#### 4.3 Pengaruh Perubahan Ukuran Blok dan *Embedding Level*

Untuk mengetahui kemampuan metode usulan lebih lanjut, pengujian metode dilakukan dengan mengubah ukuran blok dan nilai *embedding level*. Ini bertujuan untuk mengetahui pengaruh ukuran blok dan nilai *embedding level*

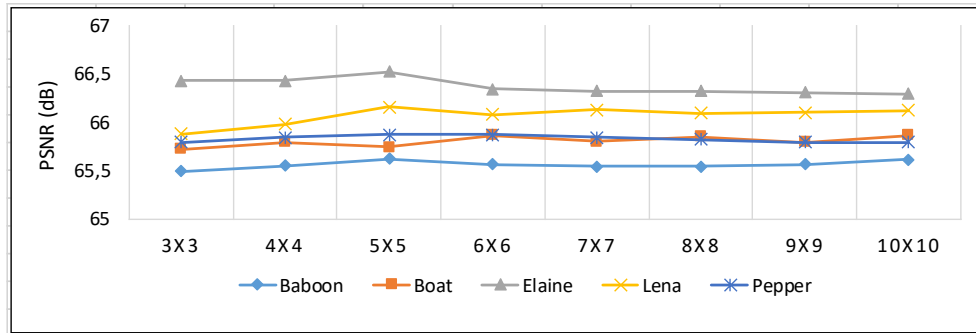
terhadap kulaitas citra *stego*. Masing-masing hasil dari pengujian tersebut dapat dilihat pada subbab dibawah ini.

#### 4.3.1 Perubahan Ukuran Blok

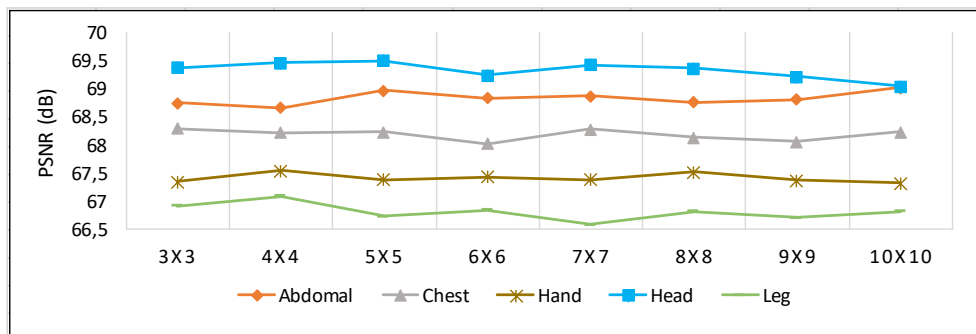
Pengujian pengaruh ukuran blok terhadap kualitas citra *stego* dilakukan dengan menggunakan *embedding level (threshold)* bernilai 1 dan data berukuran 10 Kb. Sedangkan untuk ukuran blok yang digunakan bervariasi antara 3x3-10x10. Hasil dari pengujian ini dapat dilihat pada Gambar 4.3 yang menunjukkan perubahan kualitas citra *stego* seiring dengan bertambahnya ukuran blok.

Terlihat pada gambar tersebut, kelompok citra *grayscale* rata-rata menghasilkan kualitas citra *stego* terbaik pada ukuran blok 5x5 kecuali citra ‘Boat’. Untuk kelompok citra medis, kualitas citra *stego* terbaik pada masing-masing citra dihasilkan dari blok yang berbeda-beda. Misal citra ‘Abdomal’ yang menghasilkan kualitas citra *stego* terbaik pada blok berukuran 10x10. Namun sebaliknya untuk citra ‘Hand’ blok tersebut merupakan blok yang menghasilkan citra *stego* dengan kualitas terburuk. Oleh karena itu, ukuran blok belum tentu mempengaruhi metode usulan untuk menghasilkan citra *stego* berkualitas buruk. Hal ini terjadi karena masing-masing ukuran blok menghasilkan piksel yang harus digeser (*shifted pixel*) dengan jumlah yang berbeda.

*Shifted pixel* merupakan piksel yang memiliki nilai *expanded prediction error* ( $d_{min}$  atau  $d_{max}$ ) lebih dari 1 atau kurang dari 0. Piksel ini tidak digunakan untuk penyisipan data, namun piksel ini nilainya digeser sebesar 1 poin. Misal jika awal piksel bernilai 192, maka akan digeser sebesar 1 poin menjadi 193 atau 191 berdasarkan letaknya pada *maximum part* atau *minimum part*. Contoh perbedaan jumlah *shifted pixel* pada setiap blok dapat dilihat pada Tabel 4.4 yang menunjukkan jumlah *shifted pixel* dalam citra ‘Lena’.



(a)



(b)

Gambar 4.3 Pengaruh Perubahan Blok Terhadap Kualitas Citra *Stego*,  
(a) Citra *Grayscale*, (b) Citra *Medis*

Terlihat pada tabel tersebut bahwa blok 5x5 memberikan kualitas citra *stego* terbaik dan blok berukuran 3x3 memberikan kualitas citra terburuk dibandingkan dengan ukuran blok lainnya. Kondisi ini didapatkan karena *shifted pixel* yang dihasilkan pada blok 3x3 memiliki jumlah paling banyak dan pada blok 5x5 berjumlah paling sedikit. Hal ini lah yang mempengaruhi baik atau buruknya kualitas dari citra *stego*.

Tabel 4.4 Kualitas Citra *Stego* ‘Lena’ Terhadap Perubahan Blok

Ukuran Blok	3x3	4x4	5x5	6x6	7x7	8x8	9x9	10x10
PSNR (dB)	65,88	65,98	66,16	66,08	66,13	66,09	66,1	66,12
Shifted Pixel	1863	1773	1642	1681	1639	1681	1632	1642

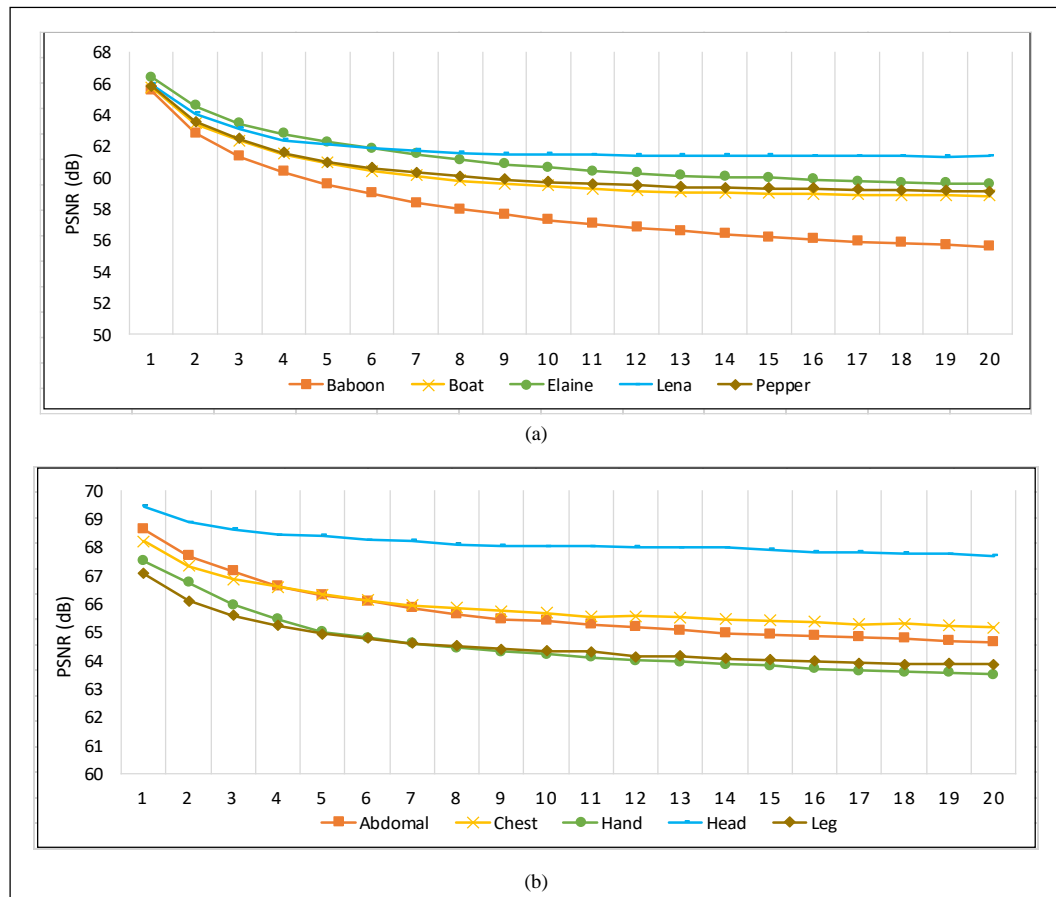
#### 4.3.2 Perubahan *Embedding Level*

Pengujian lainnya untuk mengetahui kemampuan dari metode usulan yaitu dengan mengubah nilai *embedding level*. Untuk ukuran blok dan data, pada percobaan ini masing-masing menggunakan 4x4 dan 10 Kb. Hasil dari uji coba ini dapat dilihat pada Gambar 4.4 yang menampilkan nilai PSNR untuk kelompok citra *grayscale* dan kelompok citra *medis*.

Kelompok citra *grayscale* menghasilkan citra *stego* dengan kualitas terbaik pada *embedding level* bernilai 1. Hal ini juga terjadi pada kelompok citra *medis* yang menghasilkan citra *stego* terbaik pada *embedding level* bernilai 1. Untuk kelompok *grayscale*, citra ‘Elaine’ menghasilkan kualitas citra *stego* terbaik pada *embedding level* 1-5. Sedangkan *embedding level* lainnya, citra ‘Lena’ yang menghasilkan citra *stego* terbaik. Berbeda dengan kelompok citra *medis*, citra ‘Head’ merupakan citra yang memiliki kualitas citra *stego* terbaik dibandingkan citra lainnya pada setiap nilai *embedding level*. Disisi lain, citra ‘Lena’ dan ‘Head’ memiliki *trandline* yang lebih landai dibandingkan dengan citra lainnya dalam satu kelompok. Kondisi ini terjadi karena pada kedua citra tersebut memiliki kelompok piksel *smooth* yang lebih banyak dibandingkan citra lain. Semakin banyak jumlah kelompok piksel *smooth* yang terbentuk, maka jumlah *shifted pixel* juga akan semakin sedikit. Perbandingan jumlah ini dalam citra ‘Lena’ dan ‘Head’ dapat dilihat pada Tabel 4.5.

Tabel ini menunjukkan perbandingan PSNR dan *shifted pixel* pada citra ‘Elaine’, ‘Lena’, ‘Chest’ dan ‘Head’. Jika dilihat lebih seksama, jumlah *shifted pixel* citra ‘Elaine’ pada *embedding level* 1-5 lebih sedikit dibandingkan dengan *shifted pixel* citra ‘Lena’. Ini menandakan pada *embedding level* tersebut, citra ‘Elaine’

membentuk kelompok piksel *smooth* lebih banyak dibandingkan dengan citra ‘Lena’. Namun jika dilihat *embedding level* secara keseluruhan, citra ‘Lena’ menghasilkan *shifted pixel* yang lebih stabil. Sehingga nilai PSNR citra *stego* yang dihasilkan juga tidak mengalami perubahan yang terlalu signifikan dibandingkan dengan citra *stego* dari ‘Elaine’.



Gambar 4.4 Pengaruh *Embedding Level* Terhadap Kualitas Citra Stego,  
(a) Citra *Grayscale*, (b) Citra Medis

Sama halnya dengan citra ‘Lena’, citra ‘Head’ juga menghasilkan kelompok piksel *smooth* lebih banyak yang ditandai dengan jumlah *shifted pixel* lebih sedikit dibandingkan dengan citra *stego* dari ‘Chest’. Hal ini menandakan

bahwa pada citra ‘Head’ memiliki tingkat *smoothness* lebih tinggi dibandingkan dengan citra ‘Chest’.

Tabel 4.5 Kualitas Citra *Stego* dan Jumlah *Shifted Pixel* pada Beberapa *Carrier*

<i>Embedding Level</i>		1	3	5	7	9	11	13	15
<b>PSNR (dB)</b>	<b>Elaine</b>	66,43	63,46	62,28	61,52	60,87	60,41	60,15	60,01
	<b>Lena</b>	65,98	63,12	62,14	61,76	61,51	61,47	61,43	61,43
	<b>Chest</b>	68,22	66,87	66,33	65,96	65,76	65,56	65,54	65,41
	<b>Head</b>	69,46	68,66	68,42	68,25	68,06	68,05	68	67,93
<b><i>Shifted Pixel</i></b>	<b>Elaine</b>	1334	5127	7564	9462	11415	12968	13923	14428
	<b>Lena</b>	1773	5772	7915	8815	9495	9651	9712	9759
	<b>Chest</b>	707	1676	2162	2524	2735	2915	3005	3148
	<b>Head</b>	495	996	1173	1310	1410	1476	1503	1540

Oleh karena itu dapat dikatakan bahwa, perubahan kualitas citra *stego* dipengaruhi oleh jumlah *shifted pixel* yang terbentuk. Perubahan ukuran blok tidak akan memperburuk kualitas dari citra *stego* jika dalam perubahan ukuran tersebut dapat menurunkan jumlah *shifted pixel*. Kondisi ini juga sama pada saat melakukan pengujian untuk mengubah nilai *embedding level*. Semakin tinggi jumlah *shifted pixel* dari meningkatkan nilai *embedding level* akan menurunkan kualitas citra *stego*. Kondisi ini sesuai dengan pernyataan yang disampaikan oleh Li et al (2013) bahwa semakin sedikit jumlah *shifted pixel* yang terbentuk maka kualitas citra *stego* akan semakin baik.

## **BAB 5**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil uji coba dari metode usulan memberikan beberapa kesimpulan yaitu :

1. Metode pengelompokan piksel bernilai mirip atau *adaptive pixel value grouping (APVG)* berhasil mengelompokkan piksel menjadi satu sesuai dengan *embedding level* yang ditentukan sebelumnya.
2. Metode reduksi *error expansion* (REE) berhasil memperbaiki kualitas citra *stego* yang ditandai dengan meningkatnya nilai PSNR rata-rata sebesar 4,257 dB.
3. Metode pengelompokan piksel berhasil memanfaatkan penggunaan blok-blok pada citra *carrier* dengan membentuknya menjadi kelompok bertipe *smooth*, *rough* dan *other*. Hal ini memberikan dampak pada kapasitas penyisipan yang meningkat rata-rata sebesar 156.776 bit dan diiringi penurunan kualitas citra *stego* dengan rata-rata 8,317 dB.
4. Perubahan ukuran blok tidak mempengaruhi kualitas citra *stego* jika perubahan ukuran tersebut tidak menambah jumlah *shifted pixel*.
5. Nilai *embedding level* yang bertambah besar akan menurunkan kualitas citra *stego* karena mempengaruhi jumlah *shifted pixel* yang terbentuk.

#### **5.2 Saran**

Kekurangan dari penelitian ini adalah metode usulan belum mampu mencegah perubahan susunan indeks sebelum penyisipan dan sesudah penyisipan. Oleh karena itu, penelitian selanjutnya diharapkan dapat mengatasi masalah ini. Saran pengembangan selanjutnya adalah proses *sorting* dapat dilakukan sebelum proses pengelompokkan untuk mengetahui tingkat perubahan jumlah *shifted pixel*. Disisi lain, penelitian yang akan datang disarankan untuk mengembangkan metode usulan ini agar dapat dilakukan penyisipan data dengan konsep *multilayer*

*embedding* dan dapat melakukan penghitungan tingkat *smoothness* sebuah blok yang lebih baik. Hal lain yang perlu diperhatikan adalah penelitian ini memiliki jenis *location map* yang terlalu banyak, dan konsep reduksi yang belum sempurna yaitu hanya mereduksi piksel yang disisipi sedangkan untuk piksel yang digeser (*shifted*) tidak dilakukan apa-apa. Kondisi ini diharapkan dapat diselesaikan pada penelitian selanjutnya.



## DAFTAR PUSTAKA

- Al Huti, M. H. A., Ahmad, T. & Djanali, S., 2015. *Increasing the capacity of the secret data using DE pixels block and adjusted RDE-based on Grayscale Image*. Surabaya, International Conference on Information & Communication Technology and Systems.
- Al-Ani, Z. K., Zaidan, A. A., Zaidan, B. B. & Alazani, H. O., 2010. Overview: Main Fundamentals for Steganography. *Journal of Computing*, 2(3), pp. 158-165.
- Al-Najjar, Y. A. Y. & Soong, D. C., 2012. Comparison of Image Quality Assessment: PSNR, HVS, SSIM, UIQI. *International Journal of Scientific & Engineering Research*, 3(8), pp. 1-5.
- Al-Qershi, O. M. & Khoo, B. E., 2011. High Capacity Data Hiding Schemes for Medical Images based on Difference Expansion. *Journal of Systems and Software*, 84(1), pp. 105-112.
- Andra, M. B., Ahmad, T. & Usagawa, T., 2017. Medical Record protection with improved GRDE data hiding method on audio files. *Engineering Letters*, 25(2), pp. 112-124.
- Artz, D., 2001. Digital Steganography: Hiding Data within Data. *IEEE Internet Computing*, 5(3), pp. 75-80.
- Arya, M. S., Rani, M. & Bedi, C. S., 2016. Improved Capacity Image Steganography Algorithm using 16-Pixel Differencing with n-bit LSB Substitution for RGB Images. *International Journal of Electrical and Computer Engineering*, 6(6), pp. 2735-2741.
- Bender, W., Gruhl, D., Morimoto, N. & Lu, A., 1996. Techniques for data hiding. *IBM Systems Journal*, 35(3.4), pp. 313-336.
- Cisco, 2017. *Cisco Visual Networking Index: Forecast and Methodology, 2016-2021*, s.l.: Cisco Public.
- Dragoi, I.-C. & Coltuc, D., 2014. Local-Prediction-Based Difference Expansion Reversible Watermarking. *IEEE Transactions on Image Processing*, 23(4), pp. 1779-1790.
- Gui, X., Li, X. & Yang, B., 2012. *A Novel Integer Transform for Efficient Reversible Watermarking*. Tsukuba, s.n., p. 947-950.
- Hu, Y., Lee, H.-K. & Li, J., 2009. DE-Based Reversible Data Hiding With Improved Overflow Location Map. *IEEE Transactions on Circuits and System for Video Technology*, 19(2), pp. 250-260.
- Kim, H. J. et al., 2008. A Novel Difference Expansion Transform for Reversible Embedding. *IEEE Transactions on Information Forensics and Security*, 3(3), pp. 456-465.
- Lahoulu, A., Bouridane, A., Viennet, E. & Haddadi, M., 2013. Full-Reference Image Quality Metrics Performance Evaluation Over Image Quality Databases. *Arabian Journal for Science And Engineering*, 38(9), pp. 2327-2356.

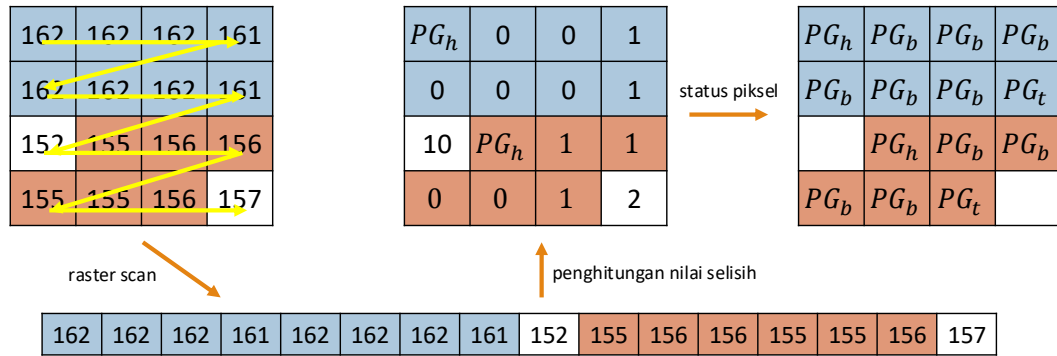
- Liu, M. et al., 2012. Reducing Location Map in Prediction-based Difference Expansion for Reversible Image Data Embedding. *Signal Processing*, 92(3), pp. 819-828.
- Li, X., Li, J., Li, B. & Yang, B., 2013. High-Fidelity Reversible Data Hiding Scheme Based On Pixel-Value-Ordering and Prediction-Error-Expansion. *Signal Processing*, Volume 93, pp. 198-205.
- Li, X., Yang, B. & Zeng, T., 2011. Efficient Reversible Watermarking Based on Adaptive Prediction-Error Expansion and Pixel Selection. *IEEE Transaction on Image Processing*, 20(12), pp. 3524-3533.
- Mukhopadhyay, J., 2011. *Image and Video Processing in The Compressed Domain*. 1st penyunt. Boca Raton: CRC Press.
- Ni, Z., Shi, Y.-Q., Ansari, N. & Su, W., 2006. Reversible Data Hiding. *IEEE Transaction on Circuits and Systems for Video Technology*, 16(3), pp. 354-362.
- Peng, F., Li, X. & Yang, B., 2012. Adaptiave Reversible Data Hiding Scheme Based on Integer Transform. *Signal Processing*, 92(1), pp. 54-62.
- Peng, F., Li, X. & Yang, B., 2014. Improved PVO-based Reversible Data Hiding. *Digital Signal Processing*, Volume 25, pp. 255-256.
- Rad, R. M., Wong, K. & Guo, J.-M., 2016. Reversible data hiding by adaptive group modification on histogram of prediction errors. *Signal Processing*, Volume 125, pp. 315-328.
- Sachnev, V. et al., 2009. Reversible Watermarking Algorithm Using Sorting and Prediction. *IEEE Transactions on Circuits and System for Video Technology*, July, 19(7), pp. 989-999.
- Simmons, G. J., 1984. The Prisoner's Problem and The Subliminal Channel. Dalam: D. Chaum, penyunt. *Advances in Cryptology: Proceedings of CRYPTO*. New York: Plenum Press, pp. 51-67.
- System Partners Healthcare Inc., 2002. *Partners Infectious Disaese Images eMicrobes Digital Library*. [Online]  
Available at: <http://www.idimages.org/>  
[Diakses 1 November 2017].
- Thodi, D. M. & Rodriguez, J. J., 2007. Expansion Embedding Technique for Reversible Watermarking. *IEEE Transactions on Image Processing*, 16(3), pp. 721-730.
- Tian, J., 2003. Reversible Data Embedding Using A Difference Expansion. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(8), pp. 890-896.
- USC-SIPI, 1977. *The USC-SIPI Image Database*. [Online]  
Available at: <http://sipi.usc.edu/database>  
[Diakses 26 April 2017].
- Wang, X., Ding, J. & Pei, Q., 2015. A Novel Reversible Image Data Hiding Scheme Based on Pixel Value Ordering and Dynamic Pixel Block Partition. *Information Sciences*, Volume 310, pp. 16-35.

- Weng, S., Pan, J.-S. & Li, L., 2016. Reversible data hiding based on an adaptive pixel-embedding strategy and two-layer embedding. *Information Sciences*, Volume 369, pp. 144-159.
- Wu, M., 2002. Part-I Fundamental Issues and Solutions. *IEEE Transaction on Image Processing*, Volume 12, pp. 685-695.
- Wu, M., Yu, H. & Liu, B., 2003. Data Hiding in Image and Video: Part II-Design and Applications. *IEEE Transactions on Image Processing*, 12(6), pp. 696-705.
- Yang, Y., Zhang, W., Liang, D. & Yu, N., 2016. Reversible Data Hiding in Medical Images with Enhanced Contrast in Texture Area. *Digital Signal Processing*, Volume 52, pp. 13-24.

*[Halaman ini sengaja dikosongkan]*

## LAMPIRAN

### Lampiran 1. Contoh *Adaptive Pixel Value Grouping (APVG)*

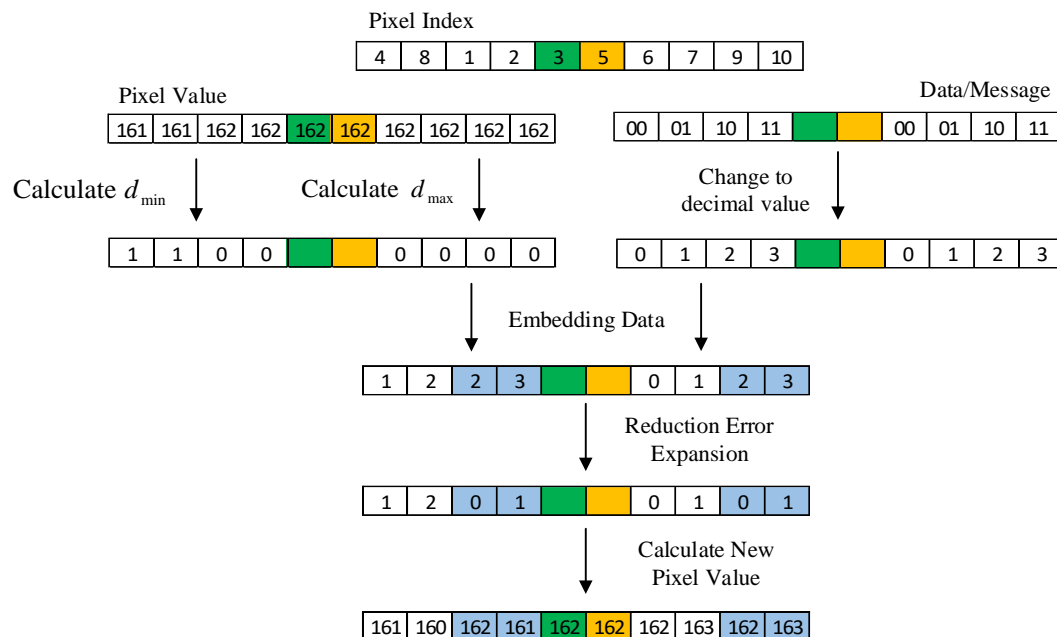


Keterangan :

*Embedding Level (EL)/Threshold* = 1

Ukuran Blok = 4

### Lampiran 2. Contoh Proses Penyisipan Data

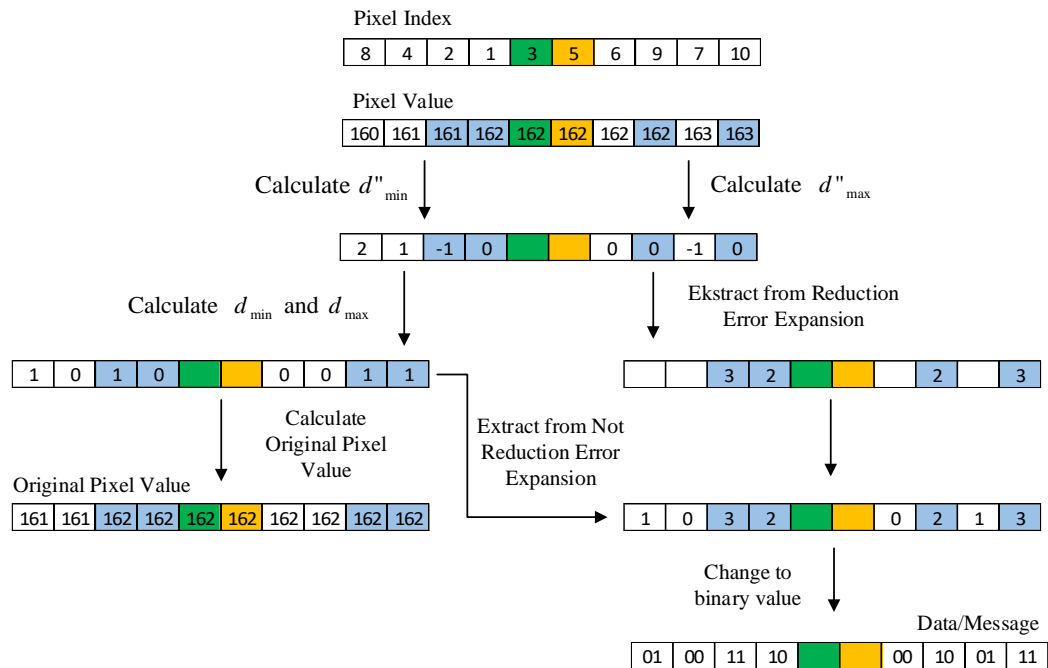


Keterangan :

: Piksel prediktor

: Piksel yang mengalami *reduksi error expansion*

### Lampiran 3. Contoh Proses Ekstraksi Data dan Pengembalian Nilai Pixel



Keterangan :

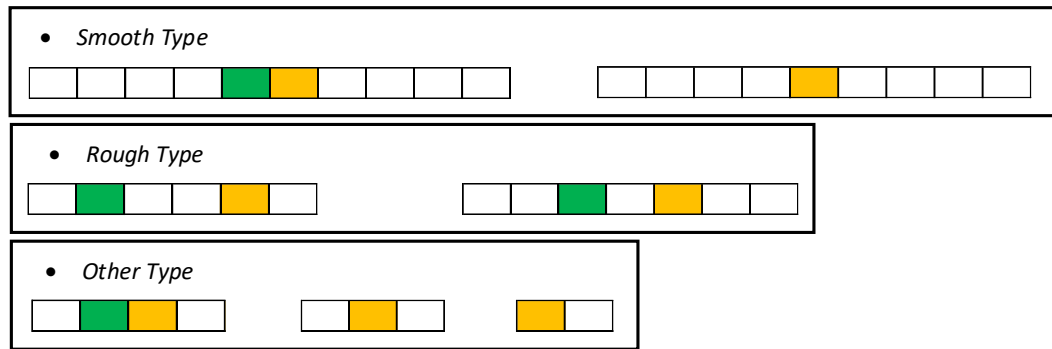


: Pixel prediktor





: Pixel yang mengalami *reduksi error expansion*

#### Lampiran 4. Jenis Kelompok Pikel



Keterangan :

  : Pikel prediktor

## BIODATA PENULIS



Hendro Eko Prabowo, biasa dipanggil Hendro dilahirkan di Rembang, kota kecil yang terletak di ujung timur provinsi Jawa Tengah dan sebagai kota perbatasan antara Jawa Tengah (Rembang) dan Jawa Timur (Tuban) 24 tahun yang lalu. Putera dari Bapak Pelda (Purn) TNI Tarmani dan Ibu Siti Rupi'ah S.Pd. dan merupakan anak pertama dari 2 bersaudara. Adik penulis bernama Guntur

Yudha Mahendra yang saat ini masih menempuh pendidikan di SMA N 1 Rembang.

Penulis menempuh pendidikan dasar di SD Negeri Mondoteko Rembang (1999-2005), kemudian melanjutkan pendidikan menengah di SMP Negeri 2 Rembang (2005-2008) dan SMA Negeri 1 Rembang (2008-2011). Gelar strata I dia dapatkan setelah menempuh pendidikan di Program Studi Pendidikan Teknik Informatika dan Komputer (PTIK) Universitas Negeri Semarang (2011-2015) dengan bidang keahlian keamanan jaringan dan komputasi bergerak.

Selama menempuh pendidikan sarjana, penulis aktif dalam kegiatan organisasi Himpunan Mahasiswa Profesi Teknik Elektro (HIMPRO TE UNNES) dan Dewan Perwakilan Mahasiswa Fakultas Teknik (DPM FT UNNES). Selain itu, penulis juga menjadi anggota *research study club* di PTIK yang menjadi landasan dasar dan motivasi penulis untuk menempuh pendidikan pasca sarjana. Untuk informasi lebih lanjut mengenai penulis atau penelitian ini dapat menghubungi email [hendro.prabowo15@gmail.com](mailto:hendro.prabowo15@gmail.com).